UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

RECONOCIMIENTO DE LOCUTOR DEPENDIENTE DE TEXTO MEDIANTE ADAPTACIÓN DE MODELOS OCULTOS DE MARKOV FONÉTICOS

-PROYECTO FIN DE CARRERA-

Cristina Esteve Elizalde
Julio de 2007

ACTA DE EXAMEN

NOMBRE DEL ESTUDIANTE:

Cristina Esteve Elizalde

TÍTULO DEL PROYECTO:

Reconocimiento de locutor dependiente de texto mediante adaptación de modelos ocultos de Markov fonéticos.

NOMBRE DEL TUTOR:

Doroteo Torre Toledano

NOMBRE DE LOS MIEMBROS DEL TRIBUNAL:

Presidente: Joaquín González Rodríguez

Vocal: Kostadin Koroutchev

Secretario: Doroteo Torre Toledano

Presidente suplente: Javier Ortega García

Vocal suplente: Alejandro Sierra Urrecho

FECHA DE LECTURA Y DEFENSA:

Madrid, a de de 2007

CALIFICACIÓN OBTENIDA:

RECONOCIMIENTO DE LOCUTOR DEPENDIENTE DE TEXTO MEDIANTE ADAPTACIÓN DE MODELOS OCULTOS DE MARKOV FONÉTICOS

AUTOR: Cristina Esteve Elizalde
TUTOR: Doroteo Torre Toledano

Área de Tratamiento de Voz y Señales
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2007

PALABRAS CLAVE

Reconocimiento locutor dependiente de texto, Verificación de locutor, Modelos Ocultos de Markov, adaptación al locutor MLLR, YOHO, BIOSEC Baseline.

RESUMEN

En este proyecto se estudian los sistemas de reconocimiento de locutor dependiente de texto, centrándonos en aquellos sistemas basados en Modelos Ocultos de Markov (HMMs). Se construye desde cero un reconocedor de locutor dependiente de texto mediante Modelos Ocultos de Markov sobre dos bases de datos. La primera de ellas y sobre la que más experimentos realizamos, por ser la base de datos de referencia en reconocimiento de locutor dependiente de texto, es la base de datos YOHO (inglés). Además, exportamos nuestro sistema a la base de datos BIOSEC Baseline (español).
En la parte de pruebas, ensayamos diferentes configuraciones de nuestro reconocedor de locutor. Primero comparamos dos técnicas de entrenamiento para ver cuál de ellas obtiene mejores resultados, como son el método tradicional de reentrenamiento Baum-Welch de los modelos independientes de locutor con voz del conjunto de locutores y la adaptación de locutores MLLR (Maximum Likelihood Linear Regression). Esta prueba la realizamos para diferentes cantidades de datos de entrenamiento (6, 24 y 96 locuciones). Además, estudiamos el efecto que produce en los resultados si variamos el número de Gaussianas por estado, así como las clases de regresión al realizar la adaptación MLLR. Por último, se analizan distintas formas de normalizar los resultados y se propone una normalización novedosa que finalmente se aplica con éxito.
Para finalizar, se presentan las conclusiones y se proponen líneas de trabajo futuras.

ABSTRACT

In the present project, text-dependent speaker recognition systems have been studied, focusing on systems implemented by Hidden Markov Models. A HMM-based text-dependent speaker recogniser was built from scratch and applied to two databases. The first one, on which we obtain most of our results, is the reference database in system assessment, is the well known YOHO database (recorded in english). Moreover, we tested our system on the database called BIOSEC Baseline (in spanish).

In the experimental part of the project, we try out various configurations for our recognition system. First of all, we make a comparison between two training techniques in order to determine which one achieves better results, the common method of independent speaker models retraining known as Baum-Welch reestimation and MLLR (Maximum Likelihood Linear Regression) speaker models adaptation. This test was performed for different amounts of enrolment speech (6, 24 and 96 utterances). Furthermore, we have tested the outcome obtained by changing the number of mixtures per state, as well as the number of regression classes when we apply MLLR adaptation. Lastly, we analyse novel different ways of score normalization and these techniques are successfully implemented.
Finally, conclusions are drawn, and future lines of work are proposed.

Agradecimientos

El presente proyecto de investigación no hubiera sido posible sin la valiosa cooperación de las personas por las que siento un profundo agradecimiento. Quisiera destacar en primer lugar a mi tutor, el profesor Doroteo Torre Toledano por su dedicación y su experiencia profesional, a Joaquín González y al resto de miembros del grupo ATVS por su colaboración y apoyo, a mis compañeros por su amistad, y finalmente a mis padres por su cariño.

Cristina Esteve Elizalde
Julio de 2007

Este proyecto ha sido realizado en el Área de Tratamiento de Voz y Señales (ATVS) en la Escuela Politécnica superior de la Universidad Autónoma de Madrid. El proyecto ha sido financiado parcialmente por el Ministerio de Defensa y el Ministerio de Educación y Ciencia a través del proyecto TEC2006-13170-C02-01.

Índice

Índice de Tablas

Índice de Figuras

Glosario

Autenticar
En Biometría, la palabra "autenticación" suele usarse como sinónimo genérico de "verificación".
Base de datos
Recopilación de uno o más archivos computarizados. En el caso de sistemas biométricos, estos archivos pueden ser lecturas del sensor biométrico, plantillas, resultados de coincidencias, información sobre el usuario final, etc.
Biometría
Parte de la biología que estudia cuantitativamente la variabilidad individual de los seres vivos utilizando métodos estadísticos.
Captura
Proceso de recopilación de una muestra biométrica de un individuo por medio de un sensor.
Características
Características matemáticas distintivas derivadas de una muestra biométrica, utilizadas para generar una referencia.
Comparación
Proceso de comparación de una referencia biométrica con una referencia o referencias almacenadas con anterioridad, para tomar una decisión sobre identificación o verificación.
Curva DET (compensación por error de detección)
Trazo gráfico de las tasas de error medidas. Por lo general, las curvas DET trazan las tasas de error de decisión (tasa de falso rechazo vs. tasa de falsa aceptación).
Dependiente de texto
Un sistema de verificación de locutores en el que el texto que el locutor debe decir es conocido por el sistema.
Decisión
Acción a seguir (automática o manual) que resulta de la comparación de la puntuación obtenida por el usuario con la escala del sistema.
DTW (Dynamic Time Warping)
Alineamiento Temporal Dinámico.
Enfrentamiento
Proceso que incluye la comparación de una muestra biométrica con una plantilla almacenada anteriormente, y el cálculo del grado de semejanza. Los sistemas toman las decisiones basándose en este resultado y en la relación (por encima o por debajo) con la escala predeterminada.
Escala Mel
La escala Mel es una aproximación a la escala perceptual humana.
Extracción
Proceso de conversión de una muestra biométrica capturada en datos biométricos para que puedan ser comparados con una referencia.
GMM (Gaussian Mixture Model)
Modelo de Mezclas Gaussianas
HMM (Hidden Markov Model)
Modelo oculto de Markov
Identificación
Tarea en la cual el sistema biométrico busca en una base de datos una referencia que

coincida con la muestra biométrica suministrada y, de encontrarla, devuelve la identidad correspondiente. Se recopila información biométrica y se la compara con todas las referencias en la base de datos.

Impostor
Persona que somete una muestra biométrica con la intención deliberada o involuntaria de declarar la identidad de otra persona en el sistema biométrico.

Independiente de texto
Un sistema de verificación de locutores en el que el sistema no conoce el texto que el locutor ha dicho.

MAP (Maximum a Posteriori)
Método de adaptación de modelos independientes de locutor a los distintos locutores.

MFCC (Mel Frequency Cepstral Coefficients)
Coeficientes cepstrales en escala de frecuencias Mel.

MLLR (Maximum Likelihood Linear Regression)
Método de adaptación de modelos independientes de locutor a los distintos locutores mediante transformaciones lineales.

Modalidad
Tipo o clase de sistema biométrico. Por ejemplo: reconocimiento de rostro, reconocimiento de huellas dactilares, reconocimiento de iris, etc.

Modelo
Representación utilizada para caracterizar a un individuo. Los sistemas biométricos basados en características de comportamiento, debido al dinamismo inherente de estas características, utilizan modelos en vez de plantillas estáticas

Muestra biométrica
Información o datos computarizados, obtenidos por medio de un dispositivo con sensor biométrico. Por ejemplo, imágenes de rostros o de huellas dactilares.

NIST (National Institute of Standards and Technology)
Organismo federal, no regulador, perteneciente a la Cámara de Comercio de los Estados Unidos que desarrolla y promueve medidas, estándares y tecnología para aumentar la productividad, facilitar el comercio y mejorar la calidad de vida.

PIN (Número de Identificación Personal)
Método de seguridad utilizado para mostrar “lo que uno sabe”. Según cada sistema, la clave PIN puede ser utilizada para declarar una identidad o para verificar una identidad antes declarada.

Plantilla
Representación digital de las características distintivas de un individuo, que contiene la información extraída de una muestra biométrica. Las plantillas se utilizan durante la autenticación biométrica como base de comparación.

Reconocimiento del habla
Tecnología que permite que una máquina reconozca las palabras pronunciadas. El reconocimiento del habla no es una tecnología biométrica.

Reconocimiento de locutor
Modalidad biométrica que utiliza el habla de una persona, una característica influenciada tanto por la estructura física del tracto vocal del individuo como por las características de comportamiento del individuo, para fines de reconocimiento. Se divide en identificación y verificación de locutor.

Reestimación Baum-Welch
Método de entrenamiento de un Modelo Oculto de Markov.

Registro

Proceso de recopilación de muestra biométrica de un usuario final, conversión de la misma en referencia biométrica y almacenamiento de la referencia en la base de datos del sistema biométrico para posterior comparación.

ROC (Característica de funcionamiento del receptor)
Método para mostrar el rendimiento de precisión medida de un sistema biométrico. La característica ROC en una verificación compara la tasa de falsa aceptación con la tasa de verificación.

Sistema biométrico
Componentes individuales múltiples (tales como sensor, algoritmo de coincidencia y visualización de resultado) que se combinan para crear un sistema totalmente funcional. Un sistema biométrico es un sistema automatizado capaz de:

1. capturar una muestra biométrica del usuario final;
2. extraer y procesar los datos biométricos de dicha muestra;
3. almacenar la información extraída en una base de datos;
4. comparar los datos biométricos con los datos con los modelos de referencia;y
5. decidir el grado de coincidencia e indicar si se ha logrado una identificación o verificación de identidad o no.

Un sistema biométrico puede ser parte componente de un sistema mayor.

Sistema biométrico multimodal
Sistema biométrico que emplea múltiples rasgos biométricos.

Sistema biométrico unimodal
Sistema biométrico que emplea un único rasgo biométrico.

Tasa de falsa aceptación (FAR)
Estadística utilizada para medir el rendimiento biométrico durante la tarea de verificación. Porcentaje de veces que un sistema produce una falsa aceptación, lo cual ocurre cuando un individuo es erróneamente vinculado con la información biométrica existente de otra persona.

Tasa de falso rechazo (FRR)
Estadística utilizada para medir el rendimiento biométrico durante la tarea de verificación. Porcentaje de veces que el sistema produce un falso rechazo. Ocurre un falso rechazo cuando un individuo no es vinculado con su propia plantilla biométrica existente.

Tasa de igual error (EER)
Estadística utilizada para mostrar el rendimiento biométrico; por lo general, durante la tarea de verificación. La tasa EER es la ubicación en una curva ROC o DET donde la tasa de falsa aceptación y la tasa de falso rechazo son iguales. Por lo general, cuánto más bajo sea el valor de la tasa de igual error, mayor será la precisión del sistema biométrico. Observe, sin embargo, que la mayoría de los sistemas operativos no están preparados para funcionar con la "tasa de igual error", de modo que la verdadera utilidad de esta medida está limitada a la comparación con el rendimiento del sistema biométrico.

Umbral
Valor predeterminado de un usuario para las tareas de verificación o identificación de grupo abierto en los sistemas biométricos. La aceptación o el rechazo de los datos biométricos depende de si el resultado de coincidencia se encuentra por encima o por debajo de la escala. La escala es ajustable de modo que el sistema biométrico puede ser más o menos estricto según los requisitos de cada aplicación biométrica.

Verificación
Tarea durante la cual el sistema biométrico intenta confirmar la identidad declarada de un individuo, al comparar la muestra suministrada con una o más plantillas registradas con anterioridad.

Capítulo 1

Introducción y objetivos

1 Introducción y objetivos

Tradicionalmente, la verificación de la identidad se realizaba mediante contraseñas o números personales. Sin embargo, dichos sistemas son muy vulnerables a ataques fraudulentos ya que sólo se requiere conocer la contraseña para tener acceso al sistema.

Para aumentar la seguridad de estos sistemas se han desarrollado técnicas de reconocimiento de personas basadas en rasgos biométricos, tales como la huella dactilar, el iris ocular o la voz humana, haciendo la suplantación de la identidad una tarea difícil.

En este proyecto se desea combinar un rasgo biométrico, la voz con el uso de números personales para desarrollar un reconocedor de locutor dependiente de texto en español y en inglés. Esta aproximación asocia las ventajas de las contraseñas con las de los rasgos biométricos para conseguir un más elevado nivel de seguridad.

1.1 Reconocimiento de locutor: variantes y aplicaciones

El reconocimiento automático de locutor consiste en reconocer a una persona a través de su voz sin supervisión humana.

1. Introducción y objetivos

Los sistemas de reconocimiento automático de locutor se pueden clasificar en tres tipos:

- Identificación de locutores
- Verificación de locutores
- Seguimiento y agrupamiento de locutores

En la identificación de locutor el locutor no aporta información sobre su identidad y es el sistema el que determina quién es a partir de su voz dentro de un conjunto de posibles candidatos o, si se trata de identificación en conjunto abierto, si el locutor es conocido o no por el sistema. Por el contrario, la tarea de los sistemas de verificación de locutor es determinar si el locutor es o no quién dice ser. Por último, el seguimiento y agrupamiento consiste en etiquetar qué locutor está hablando en un segmento de voz y cuándo se producen cambios de locutores.

De los tres tipos, el proyecto se va a realizar dentro del marco de verificación de locutores. Sin embargo, cada uno de éstos poseen características distintas y aplicaciones interesantes que cabe destacar:

La identificación de locutores se puede utilizar para restringir el acceso a información a personas no autorizadas. Por otro lado, el seguimiento y agrupamiento de locutores tiene su utilidad en la trascripción de noticias o reuniones, con el fin de aislar la voz de cada uno de los locutores en una grabación. La verificación de locutores tiene también numerosas aplicaciones comerciales importantes, por ejemplo las transacciones bancarias a través del teléfono. Además de ésta, existen muchas otras aplicaciones comerciales, todas destinadas a aumentar la seguridad en la verificación de la identidad como podría ser la gestión de identidad en centrales de atención al cliente, haciendo posible confirmar la identidad del usuario que llama y certificar las operaciones que realice como la contratación o baja de nuevos servicios. Otra aplicación podría ser el restringir el acceso a personas no autorizados a bases de datos con información confidencial de clientes. Muy importante también son las aplicaciones en el ámbito forense, puesto que se puede emplear en juicios para comprobar si la voz empleada como prueba coincide con la del acusado.

Por las múltiples aplicaciones potenciales mencionadas anteriormente y el interés por el tema, el proyecto se desarrollará en el ámbito de verificación de locutor.

Volviendo a la clasificación de sistemas, esta vez ya centrándonos en los sistemas de verificación de locutor, otro criterio importante gira en torno a su dependencia con el texto pronunciado, distinguiendo entre:

- Sistemas dependientes de texto y
- Sistemas independientes de texto

En los primeros, la locución de entrenamiento y la de verificación suelen ser el mismo texto. Fundamentalmente consiste en una palabra o frase clave (contraseña) que le permite el acceso al sistema al usuario. En estos sistemas, la contraseña es conocida por el sistema y suele ser fija, requiriendo un nuevo entrenamiento cada vez que se desea cambiar de contraseña. Un problema de estos sistemas es que son relativamente fáciles de atacar en caso de que el impostor grabe la palabra clave pronunciada por el usuario. Para evitar este tipo de ataques se introducen los sistemas "text-prompted" o de texto solicitado, en los que el sistema además de solicitar la contraseña al usuario solicita repetir un código o frase elegido aleatoriamente, y que por tanto evita la posibilidad de utilizar grabaciones.

Por el contrario, en los sistemas independientes de texto la locución de entrenamiento y la de test no coinciden, siendo la locución de test desconocida por el sistema. En este caso, el sistema no utiliza ningún tipo de contraseña, sino únicamente el rasgo biométrico de la voz.

Ambas tareas son distintas y emplean por ello diferentes técnicas. En sistemas independientes de texto se utilizan tradicionalmente técnicas basadas en GMMs (Gaussian Mixture Models), mientras que en sistemas independientes de texto se suelen utilizar técnicas de DTW (Dynamic Time Warping) o HMMs (Modelos Ocultos de Markov).

1.2 Objetivos

El objetivo de este proyecto es construir un sistema de verificación de locutor dependiente de texto. Éste sistema se va a evaluar sobre distintas bases de datos, una de ellas en inglés y otra en español. Se implementa mediante la técnica de modelado estadístico de los modelos ocultos de Markov (HMMs). Además se ensayan distintos métodos de entrenamiento, tales como la reestimación Baum-Welch o la adaptación

MLLR, para determinar con cuál se alcanzan mejores resultados. Finalmente se aplican normalizaciones novedosas a las puntuaciones.

1.3 Organización de la memoria

Esta memoria está organizada en 6 partes: la introducción, el estudio del estado del arte y la descripción de las tecnologías a utilizar, el diseño y desarrollo, las pruebas y resultados, las conclusiones y las referencias consultadas.

Capítulo 2

Estudio del estado del arte y tecnologías a utilizar

2.1 Funcionamiento General de un sistema de verificación de locutor

La estructura común de todos los sistemas de verificación automática de locutor consta de 2 fases:

1. Fase de entrenamiento
2. Fase de verificación

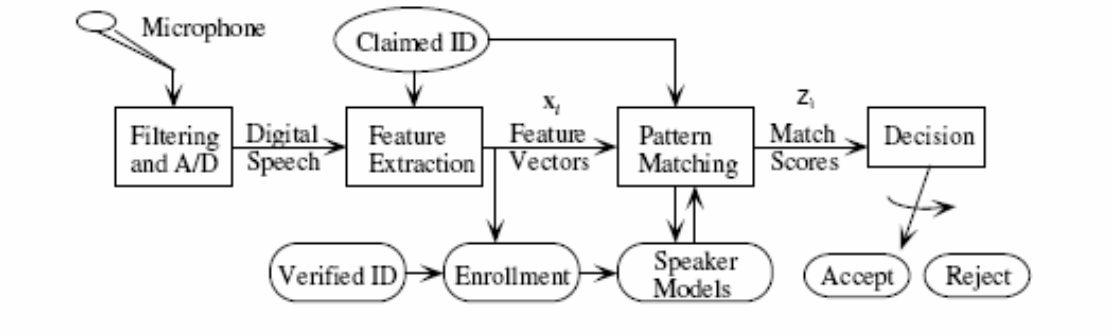

Figura 1. Sistema genérico de Verificación Automática de Locutor [Campbell, 1999].

2.1.1 Fase de entrenamiento

La primera fase corresponde a la fase de registro, en la que, a partir de una serie de locuciones pronunciadas por los futuros usuarios del sistema, se generan los modelos de referencia para cada locutor contra los que comparar en la fase de verificación. Previamente son necesarias las fases de adquisición digital de los datos y de extracción de parámetros, que se describirán en la sección 2.2 y 2.3 respectivamente. El entrenamiento de estos modelos se realiza en los HMMs y GMMs mediante distintas técnicas como la adaptación de modelos independientes del locutor a los distintos locutores o bien reentrenando los modelos con las locuciones disponibles de cada locutor. En el caso de DTW, el entrenamiento no es más que el almacenamiento de las locuciones (previamente parametrizadas) para usarlas como plantillas en la fase de verificación. Todo esto se verá más en detalle en el apartado 2.5.

2.1.2 Fase de verificación

En la fase de verificación, tras la fase previa de adquisición de la voz y la extracción de vectores de características, que se realizan exactamente igual que en la fase de entrenamiento, se realiza el cálculo de puntuaciones, enfrentando la locución a verificar al modelo del locutor que dice ser. A partir de esta puntuación y en base a un umbral, fijo o calculado para cada locutor, se toma la decisión de aceptar o rechazar al usuario en el sistema, como se explicará más adelante en el apartado 2.4.

2.2 Adquisición de voz

La adquisición de la voz se lleva a cabo mediante un micrófono o un auricular telefónico, que convierten la onda acústica en una señal analógica. A esta señal analógica se le aplica entonces un filtro antialiasing para limitar el ancho de banda de la señal a la frecuencia de Nyquist. La señal analógica se muestrea para convertirla en una señal digital con un convertidor analógico/digital. Hoy en día los convertidores A/D para aplicaciones de voz muestrean habitualmente a una tasa de 8000 hasta 44100 muestras por segundo con una resolución de 12 a 16 bits por muestra.

En aplicaciones locales de verificación de locutor, el canal analógico es simplemente el micrófono, el cable y el acondicionamiento de la señal analógica. Es por esto que la señal digital resultante puede llegar a ser de alta calidad, sin las distorsiones que se producen al transmitirse las señales por las líneas telefónicas.

2.3 Extracción de parámetros

Los Mel Frequency Cepstral Coefficients (coeficientes cepstrales en escala de frecuencias Mel) son coeficientes para la representación del habla basados en la percepción auditiva humana. Se derivan de la Transformada de Fourier (FT) y de la Transformada discreta del coseno (DCT). La diferencia básica entre FT y MFCC es que en MFCC las bandas de frecuencia están espaciadas logarítmicamente (según la escala Mel) para modelar la respuesta auditiva humana más apropiadamente que las bandas espaciadas linealmente de la FT. Esto permite un procesado de datos más eficiente, por ejemplo, en compresión de audio. La imagen siguiente representa el procesado de la señal que se realiza en un sistema típico para computar los coeficientes MFCC.

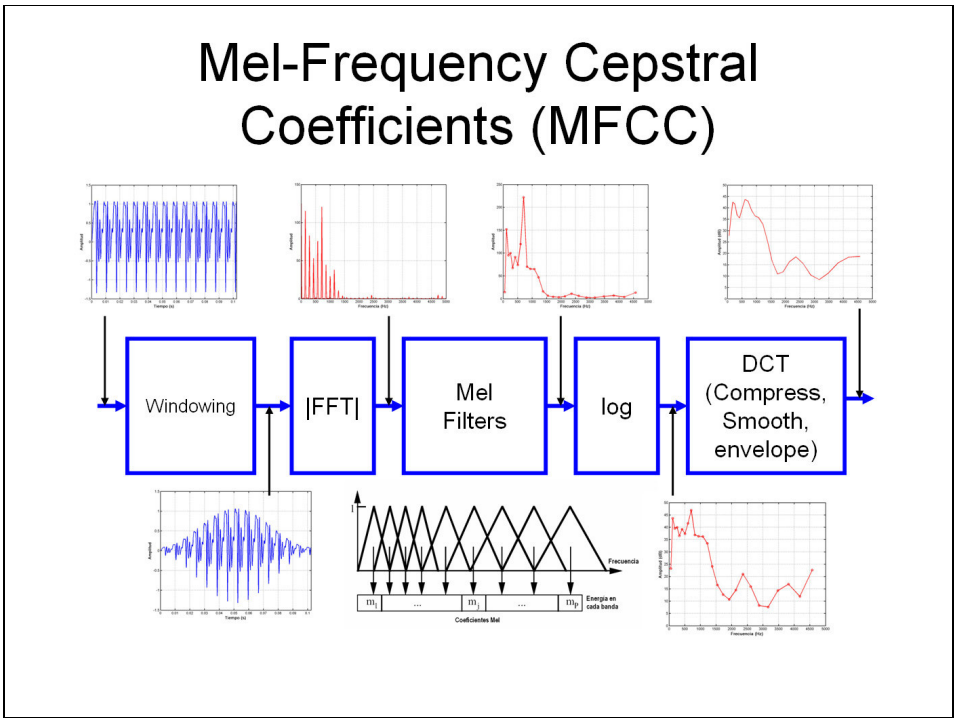

Figura 2. Diagrama de bloques del proceso de cálculo de los MFCCs

La señal acústica, muestreada a 8 KHz en el caso de señales telefónicas, se diferencia (filtro de preénfasis) y se divide en un número de segmentos solapados (enventanado), cada uno de 25 ms de longitud solapados 15 ms entre sí. A continuación la señal se filtra mediante un banco de filtros de diferentes frecuencias y amplitudes para dar más resolución en las bajas frecuencias, como ocurre en el sistema auditivo humano. Este filtrado se realiza en el dominio de la frecuencia al que se pasa calculando previamente

la FFT. De la salida de cada filtro se calcula la energía en promedio (para la ventana de 25ms) y los valores obtenidos se pueden ver como una nueva señal de tiempo discreto. Así por ejemplo, usando un banco de 40 filtros se obtiene, para cada trama de voz de 25ms, un vector de 40 coeficientes. Al trasformar esta señal a través de una DCT (Discrete Cosine Transform), se obtienen unos parámetros (de los que se toman habitualmente de 13 a 20) aproximadamente incorrelados entre ellos: estos son los coeficientes MFCC. En particular, el primer coeficiente, C_0 representa la energía de la señal y se usa o no dependiendo de la aplicación (en caso de usarlo habitualmente se normaliza para compensar variaciones de energía debidas a proximidad al micrófono u otros efectos colaterales indeseados). Aparte de estos primeros coeficientes se suelen usar también las velocidades y/o las aceleraciones, que representan la evolución temporal de los fonemas al pasar de unos a otros (Delta-MFCC y Delta-Delta-MFCC). Los coeficientes Delta representan la variación de los coeficientes MFCC alrededor del instante de tiempo considerado. Suelen, por esto, llamarse coeficientes de primera derivada o velocidad. De modo similar, los Delta-Delta se denominan de aceleración.

2.4 Toma de decisiones y evaluación

En este apartado, se describe primero el proceso de decisión que tiene lugar en los sistemas de verificación de locutor para determinar si el locutor es aceptado o rechazado por el sistema. A continuación se explica la manera en que se evalúan los errores que se cometen en la decisión.

Debemos destacar que en este punto hemos decidido saltar de modo intencionado la descripción detallada del entrenamiento y el cálculo de puntuaciones. La razón es que estos dos aspectos, que se abordarán en el apartado 2.5 son muy dependientes de la técnica empleada (DTW, HMMs, GMMs), mientras que la toma de decisiones y la evaluación es independiente de la técnica empleada.

2.4.1 Marco genérico de la toma de decisión

Dado un segmento de voz X y un locutor S, el objetivo de la verificación del locutor es determinar si S generó la locución X. Esto se puede formalizar como un test de hipótesis básico entre las siguientes hipótesis:

H_0: X fue pronunciado por el locutor S.

H_1: X no fue pronunciado por el locutor S.

La decisión, de acuerdo con el criterio de máxima verosimilitud (Maximum Likelihood, ML), se obtiene mediante el cociente de verosimilitudes que viene dado por:

$$\frac{P(X \mid H_0)}{P(X \mid H_1)} \begin{cases} \geq \vartheta & aceptar\ H_0 \\ < \vartheta & rechazar\ H_0 \end{cases}$$

donde $P(X \mid H_i)$, i=0,1 es la probabilidad de la hipótesis H_i evaluada para el segmento de voz Y. θ es el umbral de decisión para aceptar o rechazar H_0. En principio debería ser 0, pero en aplicaciones prácticas interesa ajustar dicho umbral para controlar la relación entre las probabilidades de cometer errores en los dos sentidos posibles en la decisión. Habitualmente se suele emplear el logaritmo de este cociente:

$\Lambda(X) = \log P(X \mid H_0) - \log P(X \mid H_1)$.

Por tanto, el objetivo de los sistemas de reconocimiento de locutor es encontrar métodos para calcular ambas probabilidades, $P(X \mid H_0)$ y $P(X \mid H_1)$.

Un paso crucial en la implementación del detector es el cálculo de las probabilidades P. La forma de estimar estas probabilidades depende de la aplicación. Para reconocimiento de locutor independiente de texto no existe información a priori de lo que el locutor ha dicho y en estas circunstancias la elección más acertada es el uso de GMMs. Por el contrario, para reconocimiento de locutor dependiente de texto donde sí que existe tal información, se suelen utilizar HMMs de manera que se puede incluir información temporal adicional.

2.4.2 Medidas de los errores en la decisión

En la verificación de locutores se pueden dar dos tipos distintos de errores:

1. Falso Rechazo (FR), que se produce cuando un usuario auténtico es rechazado por el sistema, y
2. Falsa Aceptación (FA), que aparece cuando un impostor es aceptado por el sistema como si fuera un usuario auténtico.

Si se observa la distribución de las puntuaciones de usuarios e impostores se puede observar que, de manera general, ambas distribuciones se solapan, lo que supone un problema para seleccionar el umbral, a partir del cual las puntuaciones serán interpretadas como pertenecientes a usuarios registrados.

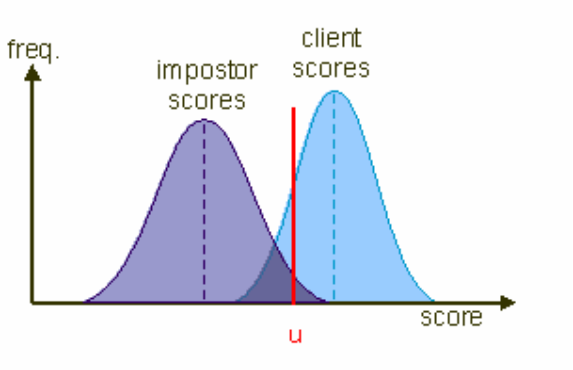

Figura 3. Distribución de usuarios e impostores, tomada de BioID

Por lo tanto, el área bajo la curva de impostores que queda por encima del umbral es la probabilidad de que un impostor sea aceptado. Esta probabilidad es la tasa de falsa aceptación (FAR o False Acceptance Rate). La probabilidad de que un usuario registrado no sea aceptado es el área bajo la curva de usuarios válidos que queda por debajo del umbral, lo que se denomina la tasa de falso rechazo (FRR False Rejection Rate).

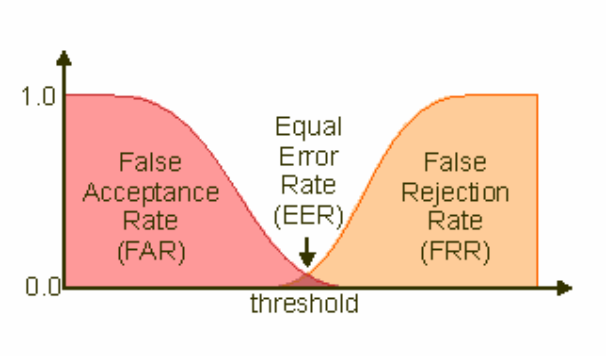

Figura 4.Curva de tasa de falsa aceptación frente a la tasa de falso rechazo. Figura tomada de BioID.

Si la distribución de puntuaciones de usuarios e impostores se solapan, la FAR y la FRR tendrán un punto de intersección, en el cual la FAR y la FRR son iguales. A este punto se le denomina tasa de equierror (Equal Error Rate ERR). Este punto se utiliza para comparar distintos sistemas y es donde el error del sistema, dado como la suma de la FAR y la FRR, se suele minimizar. Sin embargo, para poder comparar dos sistemas según el EER es necesario que éste sea calculado sobre los mismos datos de test utilizando el mismo protocolo experimental.

Como el EER no describe plenamente el rendimiento del sistema, éste se suele representar mediante las curvas ROC (Reciever Operating Curve) y las curvas DET (Detection Error Tradeoff). En ambas curvas se muestra la tasa de falsa aceptación frente a la tasa de falso rechazo para distintos niveles de umbral. Las curvas DET se obtienen a partir de las curvas ROC realizando una transformación no lineal en los ejes, de manera que las curvas no lineales de las ROC se convierten casi en rectas. Esto las hace más sencillas de analizar y comparar unas con otras, por lo que se suele optar por las curvas DET.

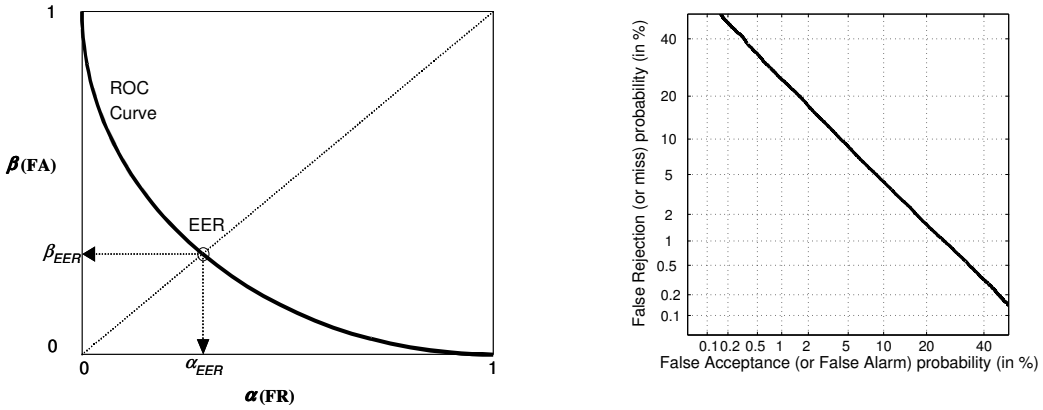

Figura 5.Ejemplo de curva ROC y curva DET

2.5 Entrenamiento y cálculo de puntuaciones

Como ya se vio, la tarea de verificación de locutor se compone esencialmente de dos fases: la fase de entrenamiento y la del cálculo de puntuaciones. Esta puntuación representa la medida de similitud entre los vectores de características del segmento de audio a verificar y un modelo de locutor. A su vez, los modelos del locutor se construyen a partir las características extraídas de uno o varios segmentos de voz de cada locutor. Cuando se desea autenticar a un usuario, se compara la señal de entrada con el modelo del locutor que dice ser y que se ha creado en la fase de entrenamiento.

Existen dos tipos de modelos: los modelos estocásticos y los modelos de plantillas (templates en inglés). En los modelos estocásticos la comparación de patrones se realiza de manera probabilística obteniendo una medida de la probabilidad condicional de la observación dado el modelo. Un ejemplo de modelado estocástico son los Modelos Ocultos de Markov (HMMs) o los modelos de mezclas de Gaussianas (GMMs).

Por el contrario, el cómputo de verosimilitudes utilizando modelos de plantillas es un proceso de comparación basado en cálculo de distancias. Se asume que la observación es una réplica no idéntica de la plantilla y se realiza un alineamiento de las secuencias observadas con las secuencias de referencia de manera que se minimice la distancia que existe entre ambas. Un ejemplo de este método es el Alineamiento Temporal Dinámico.

2.5.1 Alineamiento Temporal Dinámico (DTW)

El Alineamiento Temporal Dinámico es un método empleado en reconocimiento de locutor dependiente de texto. Esta técnica trata de compensar la variabilidad que existe entre las duraciones de los fonemas en las distintas realizaciones o pronunciaciones de una misma frase. Consiste en comparar la locución de entrada con una serie de plantillas que representan a las unidades a reconocer. El entrenamiento consiste únicamente en almacenar las distintas plantillas correspondientes a cada una de las unidades a reconocer. Las plantillas son por lo tanto un conjunto de características acústicas ordenadas en el tiempo. Para el cálculo de puntuaciones es necesario un alineamiento temporal con posibles deformaciones elásticas y una medida de distancia. Para ello se utilizan técnicas de Programación Dinámica. A continuación se describe el algoritmo para calcular esa distancia:

El objetivo es alinear de manera óptima la secuencia de vectores de parámetros de entrada $\mathbf{T} = \{t_1, t_2, ..., t_N\}$ con el modelo de referencia $\mathbf{R} = \{r_1, r_2, ..., r_M\}$, donde N es en general distinto a M debido a la variabilidad de la duración ya comentada antes. Se necesita entonces una función que relacione las N muestras de la secuencia de entrada y las M de la plantilla, minimizando la distorsión entre ambas. La función será de la forma $m = W(n)$ y debe de cumplir además las siguientes restricciones:

- $W(1) = 1$
- $W(N) = M$.

Dadas dos secuencias cualesquiera, la función W(n) es el camino de alineamiento óptimo entre ambas y se obtiene resolviendo la siguiente ecuación:

$$D^* = \min\left\{\sum_{n=1}^{N} d\left[t_n, r_{W(n)}\right]\right\}$$, donde $d[t_n, r_{W(n)}]$ es la distancia (habitualmente se trata de la distancia Euclidea) entre el instante n de la secuencia de entrada y el instante

W(n) de la plantilla. Al final del alineamiento, D^* es la distancia acumulada sobre el camino óptimo W(n) entre R y T y constituye la base para la puntuación resultante, en la que también pueden incluirse costes adicionales que penalicen caminos que sean demasiado no diagonales.

2.5.2 Modelos de Mezclas Gaussianas (GMM)

En la última década, los modelos de mezclas Gaussianas han predominado en los sistemas de reconocimiento de locutor independiente de texto. Resultan apropiados para esta tarea ya que no modelan el texto que se dice, sino que explota las características espectrales de la voz para discriminar a los locutores. El uso de GMMs en reconocimiento de locutor independiente de texto se describió por primera vez en una serie de artículos publicados por Reynolds y Rose a partir de 1990. Desde entonces, han aparecido en múltiples publicaciones de congresos y numerosos sistemas basados en GMMs han participado en las competiciones anuales NIST (National Institute of Standards and Technology).

Descripción de los GMMs:

Los sistemas de reconocimiento de locutor basados en GMMs asumen que la probabilidad condicionada que definíamos en el apartado 2.4.1 viene dada por una mezcla de distribuciones Gaussianas.

Por lo tanto, para un vector de características x de dimensión D, la densidad mezcla de Gaussianas utilizada como función de verosimilitud se define como:

$$p(x \mid \lambda) = \sum_{i=1}^{M} w_i \, p_i(x) \, .$$

Como se puede observar analizando la fórmula, se trata de una suma ponderada de las M densidades componentes siendo w_i el peso de cada de una de ellas. Cada Gaussiana viene dada a su vez por:

$$p_i(x) = \frac{1}{(2\pi)^{D/2} \left|\Sigma_i\right|^{1/2}} \exp\left\{ -\frac{1}{2} (x - \mu_i)' (\Sigma_i)^{-1} (x - \mu_i) \right\},$$

donde μ_i es el vector de medias de dimensión Dx1 y Σ_i es la matriz DxD de covarianzas.

De esta manera, cada locutor estará representado por un modelo de mezclas de Gaussianas que denotaremos $\lambda = \{w_i, \mu_i, \Sigma_i\}$ con i=1...M.

Entrenamiento de un UBM

Como veíamos en el apartado 2.4.1, es necesario contar con un modelo independiente de locutor para calcular la puntuación que se obtiene frente a este modelo y utilizarla como referencia en el test de hipótesis. El enfoque más extendido para modelar esta hipótesis alternativa es reunir locuciones de un conjunto grande de locutores que sean representativos de la población de locutores que se vayan a reconocer y entrenar con todas estas un modelo. A este modelo se le conoce como modelo universal o Universal Background Model (UBM).

No existe una medida objetiva de determinar el número apropiado de usuarios o las horas de grabación de voz para entrenar un UBM. Sin embargo, resultados empíricos demuestran, que no se experimenta ningún empeoramiento utilizando un UBM entrenado con una hora que uno entrenado con 6 horas de grabación [Reynolds et al.,2000].

Dados los datos con los que entrenar el UBM , existen varios enfoques que se pueden emplear para obtener el modelo final.

El más sencillo consiste en juntar todos los datos para entrenarlos posteriormente mediante el algoritmo EM. Es muy importante que a la hora de juntar los datos de los dos subconjuntos de la población éstos estén balanceados. Por ejemplo, si estamos generando un UBM independiente de género, debe existir un equilibrio entre las grabaciones de los hombres y de las mujeres.

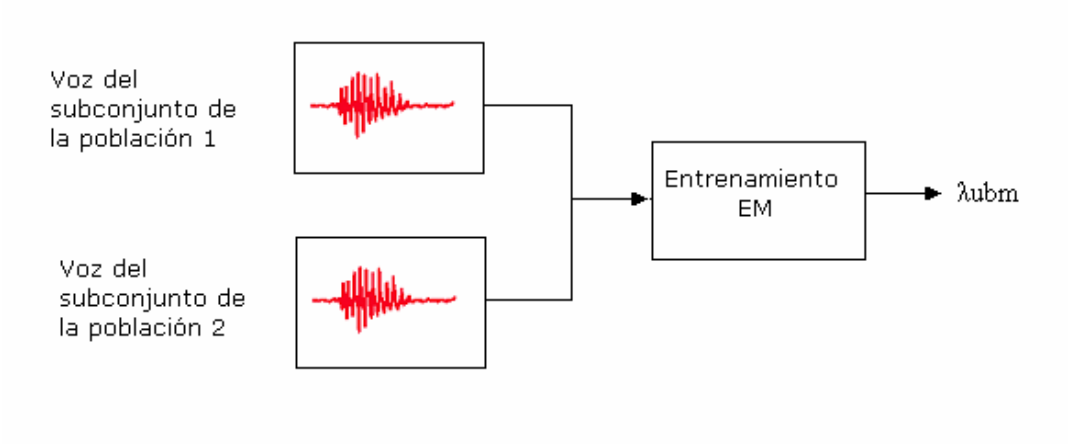

Figura 6.Primer método de generación de un UBM.

Otro método de generar el UBM consiste en entrenar de manera individual un UBM sobre cada uno de los subconjuntos, por ejemplo uno para los hombres y otro para las mujeres y después combinar los dos modelos. Este método tiene la ventaja de que permite utilizar datos no balanceados si se tiene en cuenta y se controla en el proceso final de composición del UBM.

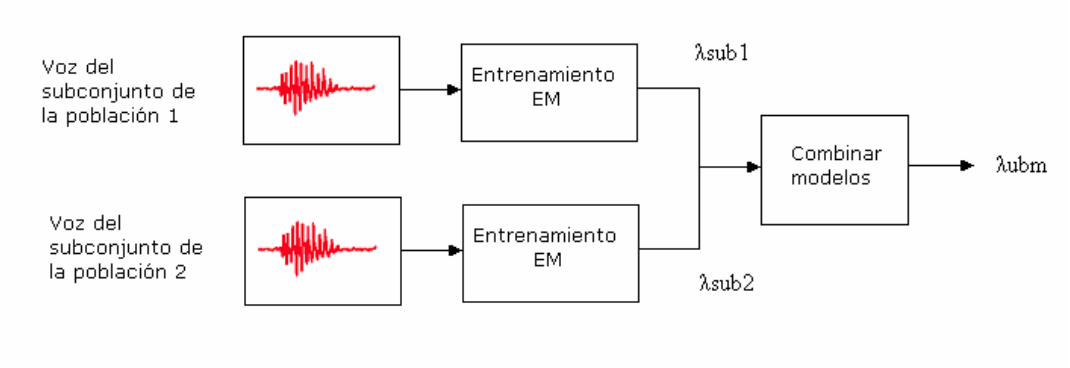

Figura 7. Segundo método de generación de un UBM.

Adaptación de los modelos del locutor

El entrenamiento de los GMMs se puede hacer de distintas maneras. La manera tradicional suele realizarse mediante estimaciones de máxima verosimilitud (Maximum Likelihood) a través del algoritmo en dos pasos estimación-maximización (Expectation Maximization), en el que de manera iterativa se refinan los parámetros del GMM para que aumente la probabilidad de generar el vector de características X dado el modelo, o lo que es lo mismo que para las iteraciones k y k+1 se cumpla que:

$$p\left(X \mid \lambda^{(k+1)}\right) \geq p\left(X \mid \lambda^{(k)}\right).$$

El otro método consiste en actualizar los parámetros del UBM mediante adaptación Bayesiana o adaptación MAP (Maximum A Posteriori). Dado que el método preferido en la actualidad es el segundo, gracias a su mayor rendimiento, es el que se va a describir en este apartado.

Al igual que el algoritmo EM, la adaptación MAP es un proceso de estimación en dos pasos. En el primer paso se estiman los estadísticos de los datos de entrenamiento para cada mezcla del UBM. En el segundo paso, se combinan estos nuevos estadísticos con los estadísticos de los parámetros del UBM.

De manera más especifica:

Dado un UBM y un vector de entrenamiento perteneciente al locutor que se desea adaptar, $X = \{x_1, x_2, ..., x_T\}$, es necesario primero determinar el alineamiento

probabilístico que existe entre el vector de entrenamiento y las mezclas que componen el UBM.

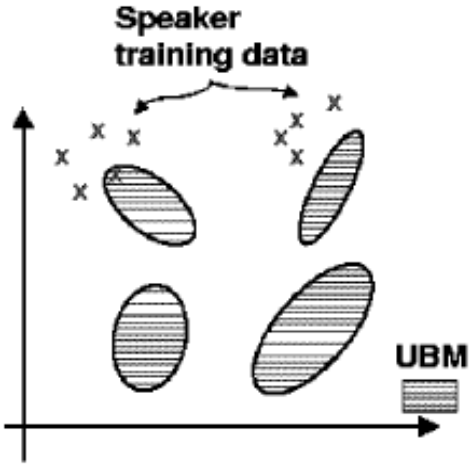

Figura 8.Ejemplo del primer paso en la adaptación MAP, tomada de [Reynolds et al.,2000].

Esto es, para la mezcla i del UBM, se calcula:

$$P(i \mid x_t) = \frac{w_i p_i(x_t)}{\sum_{j=1}^{M} w_j p_j(x_t)}$$

A partir de ese término y de x_t se calculan los estadísticos necesarios para calcular a su vez los pesos, medias y varianzas:

$$n_i = \sum_{t=1}^{T} P(i \mid x_t)$$

$$E_i(x) = \frac{1}{n_i} \sum_{t=1}^{T} P(i \mid x_t) x_t$$

$$E_i\{(x-\mu)^2\} = \frac{1}{n_i} \sum_{t=1}^{T} P(i \mid x_t) x_t^2$$

Finalmente, con estos nuevos estadísticos calculados de los datos de entrenamiento se actualizan los estadísticos antiguos del UBM para cada mezcla i para obtener los parámetros adaptados:

$$\hat{w}_i = [\alpha_i^w n_i / T + (1-\alpha_i^w) w_i] \gamma$$

$$\hat{\mu}_i = \alpha_i^m E_i(x) + (1-\alpha_i^m) \mu_i$$

$$\hat{\sigma}_i^2 = \alpha_i^v E_i(x^2) + (1-\alpha_i^v)(\sigma_i^2 + \mu_i^2) - \hat{\mu}_i^2$$

$\{\alpha_i^w, \alpha_i^m, \alpha_i^v\}$ son los coeficientes que controlan el balance entre las estimaciones antiguas y nuevas de los pesos, medias y varianzas respectivamente. Estos coeficientes se definen como sigue:

$$\alpha_i^\rho = \frac{n_i}{n_i + r^\rho}, \rho \in \{\alpha_i^w, \alpha_i^m, \alpha_i^v\},$$

siendo r^ρ un factor fijo de relevancia para el parámetro ρ.

Además, γ es un factor de escala que se calcula sobre todos los pesos adaptados para asegurar que éstos suman uno.

Este proceso se observa mejor en la figura que viene a continuación.

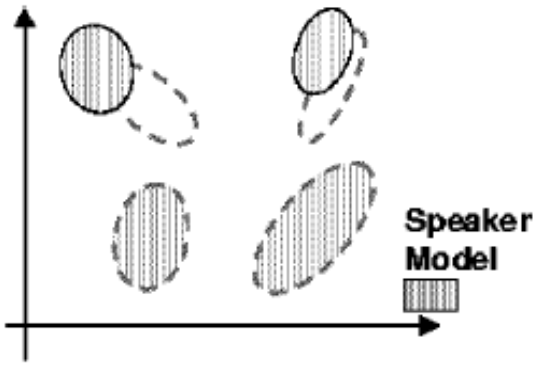

Figura 9. Segundo paso en la adaptación MAP, tomada de [Reynolds et al.,2000].

Cálculo de puntuaciones

Si volvemos al decisor del apartado 2.4.1 , veíamos que la puntuación se puede calcular como el logaritmo del cociente de verosimilitudes, quedando en el caso de un sistema GMM-UBM de la siguiente manera:

$$\Lambda(X) = \log P(X \mid \lambda_{hyp}) - \log P(X \mid \lambda_{ubm}),$$

donde λhyp es el GMM del locutor y λubm es el GMM del UBM.

Existe una manera de agilizar el proceso de puntuación (fast-scoring):

1. Para cada vector de características se determinan las C mezclas que más contribuyen a la puntuación en el UBM
2. Se enfrenta el vector frente a esas C mezclas más pesadas del modelo del locutor.

Un valor habitual para C es utilizar las 5 mezclas más pesadas.

2.5.3 Modelos Ocultos de Markov

Una técnica de modelado estocástico muy utilizada son los modelos ocultos de Markov. La teoría básica de los modelos ocultos de Markov fue introducida por primera vez por Baum y sus colaboradores en una serie de artículos entre los años 1966 y 1972.La comparación de patrones en reconocimiento de locutor dependiente de texto se realiza midiendo la verosimilitud de una observación con el modelo del locutor que dice ser.

Un HMM es una máquina de estados finita, en la que las observaciones son una función probabilística del estado, es decir, el modelo es un proceso doblemente estocástico

formado por un proceso estocástico oculto no observable directamente, que corresponde a la transiciones entre estados y un proceso estocástico observable cuya salida es la secuencia de vectores espectrales.

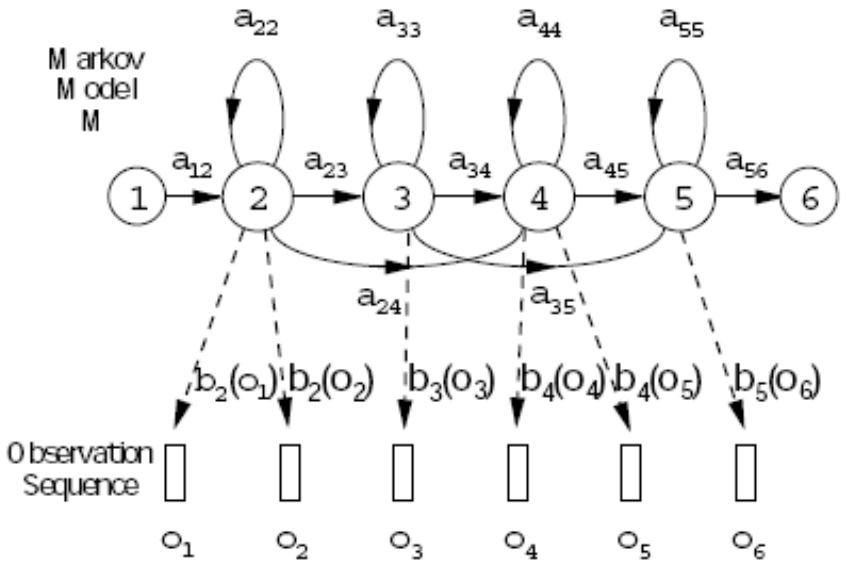

Figura 10. El modelo de generación de Markov [The HTK Book,2005]

En la Figura 10 se pueden observar los elementos que definen un HMM:

- N: el número de estados del modelo, donde q_t denota el estado en el instante de tiempo t. Los HMMs que vamos a utilizar están compuestos por 5 estados. Sin embargo en HTK, tanto el estado 1 como el estado 5 no generan ninguna salida. **$S=\{s_1,s_2,...,s_N\}$**

- La dimensión del conjunto de observaciones distintas de salida M, es decir el tamaño del alfabeto

  $\mathbf{V=\{v_1,v_2,...,v_M\}}$

- La distribución de probabilidad de transición entre estados $A=\{a_{ij}\}$:

  $\mathbf{aij=P(q_t=s_j \mid q_{t-1}=s_i)\ 1\leq i,j \leq N}$

- La distribución de probabilidades de emisión de símbolos entre estados $B = \{b_j(k)\}$:

  $\mathbf{b_j(O_k)=P(O_k \mid q_t = s_j)\ 1\leq j \leq N, 1\leq k \leq M}$ **,** donde O_k es un símbolo perteneciente a V.

- Distribución del estado inicial $\pi = \{\pi_i\}$:

  $\boldsymbol{\pi_i = P(q_0 = s_i)\ 1\leq i \leq N}$

Con todo esto, un HMM se describe como $\lambda = \{A, B, \pi\}$.

Dada esta definición surgen 3 problemas que es necesario resolver para que los HMMs tengan utilidad en aplicaciones reales.

1. Problema de evaluación de la probabilidad
2. Problema de encontrar la secuencia de estados óptima
3. El problema de entrenamiento de un modelo

Problema 1: Problema de evaluación de la probabilidad

Dada una secuencia de observación $O = \{O_1, O_2, ..., O_T\}$ y un modelo $\lambda = \{A,B,\pi\}$, ¿cómo calculamos $P(O \mid \lambda)$, la probabilidad de la secuencia de observación? Si es posible calcular esta probabilidad, entonces se podría calcular para todos los modelos y escoger aquel para el cual la probabilidad sea mayor.
La manera más directa de solucionarlo sería enumerando todas las posibles secuencias de estados de longitud T que generen la secuencia de observación O y sumando sus probabilidades según el teorema de la Probabilidad Total:

$$P(O \mid \lambda) = \sum_{Q} P(O \mid Q, \lambda) \cdot P(Q \mid \lambda) \qquad (1)$$

Para ello consideremos una determinada secuencia de estados: $Q=(q_1,q_2,...,q_T)$ donde q_1 es el estado inicial. La probabilidad de la secuencia de observación O dada la secuencia de estados Q es:

$P(O \mid Q,\lambda) = \prod_{t=1}^{T} P(O_t \mid q_t,\lambda)$, donde se asume independencia estadística de las observaciones. Por lo tanto se obtiene:

$P(O \mid Q,\lambda) = b_{q1}(O_1)\cdot b_{q2}(O_2)\cdots b_{qT}(O_T)$.

Por otra parte la probabilidad de la secuencia de estados Q se puede expresar como:

$P(Q \mid \lambda) = \pi_{q1}\cdot a_{q1q2}\cdot a_{q2q3}\cdots a_{qT-1qT}$, que se interpreta como la probabilidad del estado inicial, multiplicada por las probabilidades de transición de un estado a otro.

Sustituyendo los dos términos anteriores en el sumatorio inicial (Ecuación 1)se obtiene la probabilidad de la secuencia de observación:

$$P(O \mid \lambda) = \sum_{Q} P(O \mid Q,\lambda)\cdot P(Q \mid \lambda) = \sum_{q1,q2,...,qT} \pi_{q1}\cdot b_{q1}(O_1)\cdot a_{q1q2}\cdot b_{q2}(O_2)\cdots a_{qT-1qT}\cdot b_{qT}(O_T)$$

La interpretación del resultado obtenido es la siguiente: Inicialmente en el tiempo t=1 nos encontramos en el estado q_1 con probabilidad π_{q1} y generamos el símbolo O_1 con probabilidad $b_{q1}(O_1)$.Al avanzar el reloj al instante t=2 se produce una transición al estado q_2 con probabilidad a_{q1q2} y generamos el símbolo O_2 con probabilidad $b_{q2}(O_2)$. Este proceso se repite hasta que se produce la última transición del estado q_{T-1} al estado q_T con probabilidad a_{qT-1qT} y generamos el símbolo O_T con probabilidad $b_{qT}(O_T)$.

A pesar de haber llegado al resultado deseado se puede ver fácilmente que no es una manera muy eficiente de calcular la probabilidad ya que requiere realizar $2T\cdot N^T$ operaciones lo que por su complejidad que está en el orden de $O(N^T)$ resulta computacionalmente intratable.

Afortunadamente existe una manera más eficiente de llegar al mismo resultado. La clave está en guardar los resultados intermedios y utilizarlos para los posteriores cálculos de la secuencia de estados. A este algoritmo se le denomina el Algoritmo de Avance.

El primer paso es definir la variable hacia delante como $\alpha_t(i)=P(O_1,O_2,...,O_t, q_t = S_i \mid \lambda)$. Esta variable corresponde con la probabilidad de que el modelo λ se encuentre en el estado i habiendo generado la secuencia parcial $O_1,O_2,...,O_t$ hasta el instante de tiempo t.

$\alpha_t(i)$ se puede calcular por inducción siguiendo los siguientes pasos:

1. Inicialización:

   $\alpha_1(i)=\pi_i \cdot b_i(O_1)$, $1 \le i \le N$

   En este paso se inicializan las probabilidades hacia delante como la probabilidad conjunta del estado S_i y la observación inicial O_1.

2. Inducción:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij} \right] \cdot b_j(O_{t+1}) \ , \ 1 \le t \le T\text{-}1, \quad 1 \le j \le N$$

   La expresión entre corchetes representa la probabilidad de alcanzar el estado S_j en el instante de tiempo t+1 partiendo de todos los estados posibles S_i en el instante t habiendo observado hasta el instante t la secuencia parcial $O_1,O_2,...,O_t$. Si multiplicamos ahora dicho término por la probabilidad de observar O_{t+1} se obtiene $\alpha_{t+1}(j)$.

3. Finalización:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

   El cálculo de $P(O|\lambda)$ final se realiza sumando todas las variables hacia delante $\alpha_T(i)$ en el instante final T. Esto es así ya que por definición $\alpha_T(i)$ es igual a la probabilidad conjunta de haber observado la secuencia $O_1,O_2,...,O_T$ y encontrarnos en el estado Si: $\alpha_T(i)= P(O_1,O_2,...,O_T, q_T = S_i \mid \lambda)$, con lo que si sumamos dicha probabilidad para todos los estados posibles obtenemos la probabilidad esperada $P(O|\lambda)$.

La complejidad de este algoritmo comparado con la manera directa de calcular $P(O|\lambda)$ es mucho menor y se encuentra en el orden de $O(N^2 \cdot T)$, con lo que se el ahorro computacional es claro.

Problema 2: Problema de encontrar la secuencia de estados óptima

Decodificar un HMM consiste en encontrar la secuencia de estados óptima, dada una secuencia de observación. La resolución de este problema resulta muy importante para tareas de segmentación y reconocimiento de voz .

A diferencia del problema 1 para el que se puede dar una solución exacta, existen diferentes maneras de resolver este problema. La razón es que la definición de secuencia óptima no es única, sino que existen varios criterios de optimización.

El criterio más extendido es el que utiliza el **algoritmo de Viterbi**, que lo que trata es de encontrar la mejor secuencia de estados, es decir, maximizar la probabilidad $P(q|O,\lambda)$ o lo que es equivalente, maximizar $P(O, q|\lambda)$. En la práctica este método también se puede utilizar para evaluar HMMs.

Para encontrar la mejor secuencia de estados $Q=\{q_1,q_2,...q_T\}$ para una secuencia de observación dada $O=\{O_1,O_2,...O_T\}$ definimos la variable:

$\delta_t(i) = \max_{q1,q2,...,qt-1} P[q_1\ q_2 ...\ q_t = i, O_1\ O_2\ ...\ O_t \mid \lambda]$, que representa la secuencia de estados con mayor probabilidad en el instante t que acaba en el estado S_i y que ha generado las t primeras observaciones.

A continuación se sigue un proceso de inducción similar al algoritmo Forward-Backward ,con la excepción de que en vez de tomar la suma las probabilidades de los diferentes caminos que acaban en un mismo estado, el algoritmo de Viterbi selecciona y recuerda el mejor camino.

1. Inicialización:

   $\delta_1(i) = \pi_i \cdot b_i(O_1)$,$1 \le i \le N$

   $\phi_1(i) = 0$

   Inicialmente se define la probabilidad $\delta_1(i)$ como la probabilidad de encontrarse en el estado Si en el instante t=1 multiplicada por la probabilidad de generar el símbolo O_1.

   El vector ϕ, en el que se va a almacenar el argumento que maximiza $\delta_t(j)$ para cada valor de t y de j, toma inicialmente el valor 0.

2. Recursión:

   $\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) \cdot a_{ij}] b_j(O_t)$, $2 \le t \le T$, $1 \le j \le N$

   $\phi_t(j) = \arg\max_{1 \le i \le N} [\delta_{t-1}(i) \cdot a_{ij}]$, $2 \le t \le T$, $1 \le j \le N$

3. Finalización:

$$P^* = \max_{1 \le i \le N} [\delta_T(i)]$$

$$q_T^* = \arg\max_{1 \le i \le N} [\delta_T(i)]$$

Las iteración del punto 3 se termina cuando se han generado las T observaciones.

4. Backtracking:

$q_t^* = \phi_{t+1}(q_{t+1}^*)$,t = T – 1, T – 2, …, 1

En este último paso se reconstruye la secuencia de estados partiendo desde el estado final hasta llegar al principio.

Problema 3: Entrenamiento de un modelo

El último y más complicado de los 3 problemas plantea cómo se deben ajustar los parámetros del modelo {A, B, π } para maximizar la probabilidad de la secuencia de observación dado el modelo P(O | λ).

El principal inconveniente es que no existe ningún método analítico conocido que maximice el conjunto de parámetros a partir de los datos de entrenamiento. Se puede resolver, sin embargo, utilizando un procedimiento iterativo como el algoritmo de Baum-Welch, también conocido como el algoritmo de avance-retroceso. Este algoritmo usa los mismos principios que el algoritmo EM(Expectation Maximization).El procedimiento consiste en actualizar los pesos de forma iterativa para poder explicar mejor las secuencias de entrenamiento observadas.

Antes de describir formalmente el algoritmo de Baum- Welch es necesario definir la probabilidad hacia atrás de manera similar a como se definió la probabilidad hacia delante:

$\beta_t(i)=P(O_{t+1},O_{t+2},...,O_T, q_t = S_i \mid \lambda)$. $\beta_t(i)$ es en este caso la probabilidad de general la observación parcial $O=\{O_{t+1},O_{t+2},...,O_T\}$ desde el instante t+1 hasta el instante final T dado que el modelo se encuentra en el estado Si en el instante de tiempo t.

$\beta_t(i)$ se puede calcular por inducción como sigue:

1. Inicialización:

$\beta_T(i)=1, \ 1 \le i \le N$

2. Recursión:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j) \quad , \text{t} = \text{T} - 1, \text{T} - 2, \ldots , 1, \ 1 \leq \text{i} \leq \text{N}$$

La relación entre α y β adyacentes se puede observar mejor en la siguiente figura. α se calcula recursivamente de izquierda a derecha mientras β se calcula recursivamente de derecha a izquierda.

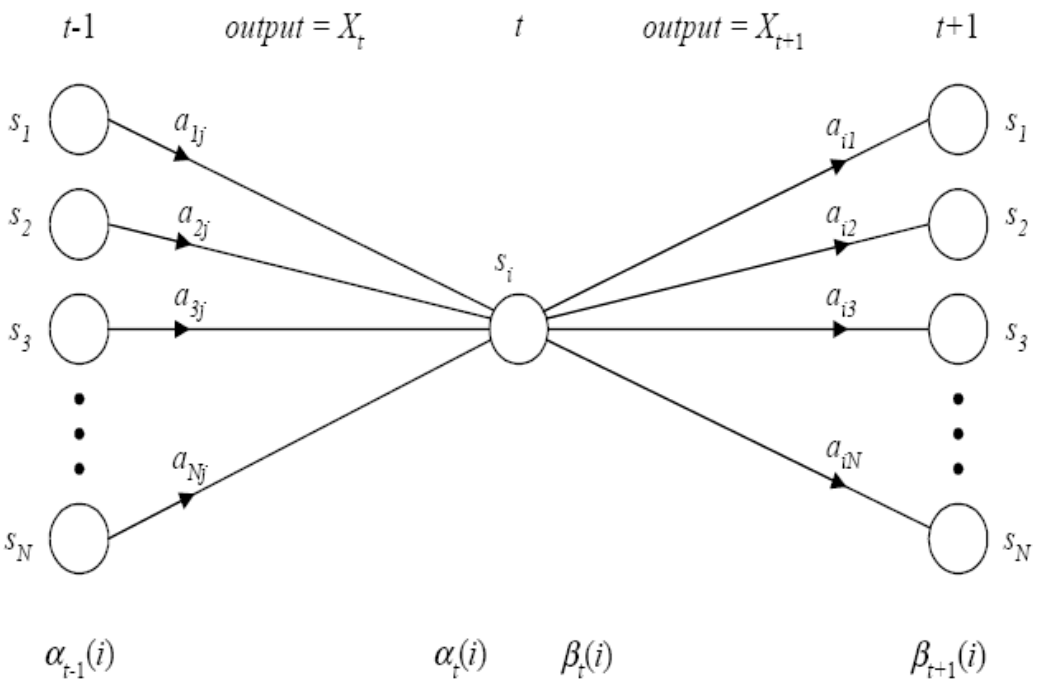

Figura 11. La relación entre α_{t-1} y α_t y β_{t-1} y β_t en el algoritmo Forward- Backward [X. Huang et al.,2001]

A continuación definimos la variable $\gamma_t(i, j)$, que representa la probabilidad de realizar una transición del estado Si al estado Sj en el instante de tiempo t dado el modelo y dada la secuencia de observación, es decir:

$$\gamma_t(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda)$$
$$= \frac{P(q_t = S_i, q_{t+1} = S_j, O \mid \lambda)}{P(O \mid \lambda)}$$
$$= \frac{\alpha_t(i) \cdot a_{ij} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j)}{\sum_{k=1}^{N} \alpha_T(k)}$$

Este resultado se puede ilustrar mejor con la siguiente figura:

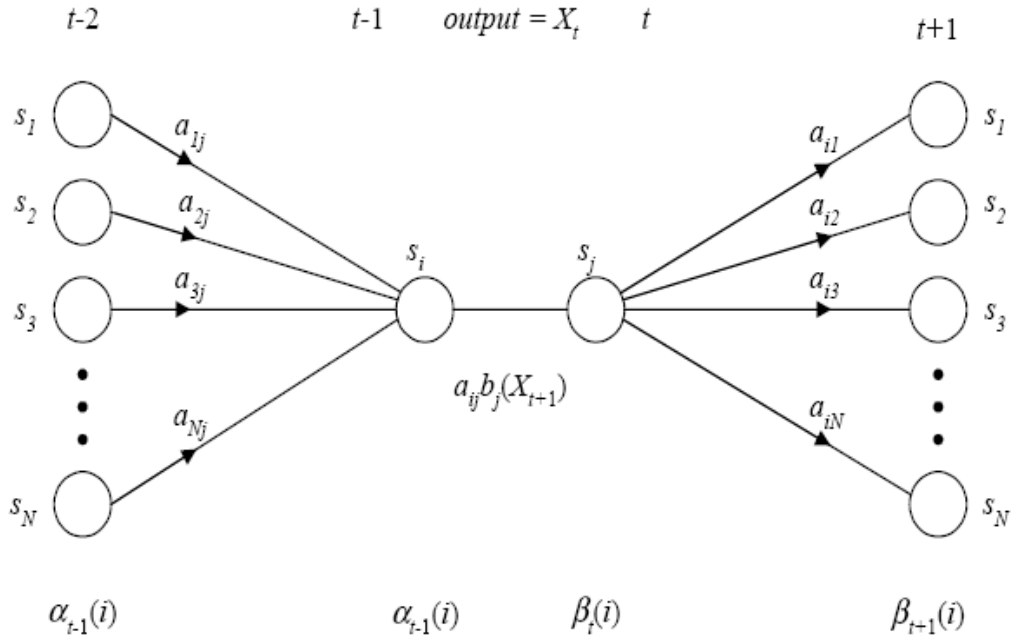

Figura 12. Ilustración de las operaciones necesarias para el cálculo de $\gamma_t(i,j)$, [X. Huang et al.,2001]

Es posible refinar iterativamente el vector de parámetros del HMM $\lambda = \{A, B, \pi\}$ si se maximiza la probabilidad de la observación ,$P(O|\lambda)$, en cada iteración. Para ello denotamos como $\hat{\lambda}$ al nuevo vector de parámetros calculado a partir del vector de parámetros λ, obtenido en la iteración anterior. De acuerdo con el algoritmo EM , esto es equivalente a maximizar la siguiente función Q:

$$Q(\lambda, \hat{\lambda}) = \sum_{S1,S2,\ldots,SN} \frac{P(O,S \mid \lambda)}{P(O \mid \lambda)} \log P(O,S \mid \hat{\lambda})$$

donde $P(O,S \mid \lambda)$ y $\log P(O,S|\hat{\lambda})$ se definen como sigue:

$$P(O,S \mid \lambda) = \prod_{t=1}^{T} a_{t-1t} b_t(O_t)$$

$$\log P(O,S \mid \hat{\lambda}) = \sum_{t=1}^{T} \log a_{t-1t} + \sum_{t=1}^{T} \log b_t(O_t)$$

Por lo tanto la ecuación inicial se puede rescribir de la siguiente manera:

$Q(\lambda,\hat{\lambda}) = Q_{ai}(\lambda,\hat{a}_i) + Q_{bj}(\lambda,\hat{b}_j)$, donde

$$Q_{ai}(\lambda,\hat{a}_i) = \sum_i \sum_j \sum_t \frac{P(O,q_{t-1}=i,q_t=j \mid \lambda)}{P(O \mid \lambda)} \log \hat{a}_{ij} \quad (1)$$

$$Q_{bj}(\lambda,\hat{b}_j) = \sum_j \sum_k \sum_{t \in o_t = V_K} \frac{P(O,q_t=j \mid \lambda)}{P(O \mid \lambda)} \log \hat{b}_j(V_k) \quad (2)$$

Como hemos separado la función en tres términos independientes, se puede maximizar Q($\lambda \mid \hat{\lambda}$) maximizando cada uno de los términos por separado, sujeto a las siguientes restricciones:

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \forall\, i$$

$$\sum_{k=1}^{M} b_j(V_k) = 1 \quad \forall\, i$$

Además, los términos en las ecuaciones 1 y 2 tienen todos las siguiente forma:

$F(x) = \sum_i y_i \log x_i$ donde $\sum_i x_i = 1$.

Haciendo uso de los multiplicadores de Lagrange, se demuestra que la función *F(x)* toma su valor máximo en $x_i = \frac{y_i}{\sum_i y_i}$.

A partir de todo esto, se obtienen las estimaciones de los parámetros del modelo HMM:

$$\hat{a}_{ij} = \frac{\frac{1}{P(O \mid \lambda)} \sum_{t=1}^{T} P(O,q_{t-1}=i,q_t=j \mid \lambda)}{\frac{1}{P(O \mid \lambda)} \sum_{t=1}^{T} P(O,q_{t-1}=i \mid \lambda)} = \frac{\sum_{t=1}^{T} \gamma_t(i,j)}{\sum_{t=1}^{T} \sum_{k=1}^{N} \gamma_t(i,k)}$$

$$\hat{b}_j(V_k) = \frac{\frac{1}{P(O \mid \lambda)} \sum_{t=1}^{T} P(O, q_t = j \mid \lambda) \cdot \delta(O_t, V_k)}{\frac{1}{P(O \mid \lambda)} \sum_{t=1}^{T} P(O, q_t = j \mid \lambda)} = \frac{\sum_{t \in O_t = V_k} \sum_{i} \gamma_t(i, j)}{\sum_{t=1}^{T} \sum_{i} \gamma_t(i, j)}$$

La probabilidad inicial $\hat{\pi}_i$ se puede derivar como un caso especial de la probabilidad de transición. Sin embargo, $\hat{\pi}_i$ se suele fijar para la mayoría de aplicaciones de voz, por ejemplo $\hat{\pi}_i = 1$ para el estado inicial.

Al observar las ecuaciones anteriores ,se puede ver que la primera corresponde con el cociente entre el número medio de transiciones del estado i al estado j y el número medio de transiciones desde el estado i.

La segunda ecuación se puede interpretar también como el cociente entre el número medio de veces que el símbolo V_k se emite desde el estado j y el número medio de veces que se emite un símbolo desde el estado j.

De acuerdo con el algoritmo EM, el algoritmo de reestimación de Baum-Welch garantiza una mejora monótona en la probabilidad en cada iteración hasta que ésta converge en un máximo local.

El algoritmo se puede resumir en los siguientes pasos:

1. **Inicialización:** Se escoge una estimación inicial del modelo λ.
2. **Paso E:** Se calcula la función auxiliar Q($\lambda, \hat{\lambda}$) a partir de λ.
3. **Paso M:** Se calcula $\hat{\lambda}$ de acuerdo con las ecuaciones de reestimación para maximizar la función auxiliar Q.
4. **Iteración:** λ pasa a tomar el valor de $\hat{\lambda}$ y se repite el algoritmo desde el paso 2 hasta que converge.

Adaptación MLLR (Maximum Likelihood Linear Regression)

La adaptación MLLR [The HTK Book, 2005] es un tipo de adaptación lineal que realiza una serie de transformaciones para reducir las diferencias entre el modelo independiente del locutor inicial y las locuciones de cada locutor que se van a emplear

en la adaptación. El efecto de estas transformaciones es desplazar las medias y modificar las varianzas del sistema inicial de manera que la probabilidad de que cada estado del sistema HMM inicial genere las locuciones de adaptación sea mayor. Las matrices de transformación se obtienen mediante la técnica EM (Expectation-Maximisation).

El nuevo vector de medias viene dado por:

$\hat{\mu} = W\xi$,donde W es la matriz de transformación de dimensiones nx(n+1) (n es la dimensión de los datos) y ξ es el vector de medias extendido

$\xi = [1 \quad \mu_1 \quad \mu_2 \quad ... \quad \mu_n]^T$.

Por lo tanto, W se puede descomponer en: $W = [b \quad A]$, siendo A una matriz de transformación nxn y b un vector de bias.

Existen dos formas de realizar la adaptación de las varianzas. La primera es:

$$\hat{\Sigma}_m = B_m^T H_m B_m ,$$

donde H_m es la transformación lineal a estimar y B_m es la inversa del factor de Choleski de Σ_m^{-1}, de manera que:

$$\Sigma_m^{-1} = C_m C_m^T$$

$$B_m = C_m^{-1}$$

Esta forma de transformación resulta en una matriz de covarianzas completa efectiva, siempre que la matriz de transformación H_m esté completa a su vez. Esto hace que el cálculo de puntuaciones sea altamente ineficiente.

La segunda manera y más eficiente de realizar la transformación de la matriz de covarianzas. Ésta se obtiene de la siguiente manera:

$$\hat{\Sigma} = H\Sigma H ,$$

donde H es la matriz de transformación de covarianzas nxn. Este tipo de transformación, se puede implementar de manera eficiente como una transformación de las medias y del vector de características:

$$N(o;\mu, H\Sigma H) = \frac{1}{|H|^2} N(H^{-1}o; H^{-1}\mu, \Sigma) = |A|^2 N(Ao; A\mu, \Sigma),$$

siendo $A = H^{-1}$. Utilizando esta forma es posible estimar y aplicar transformaciones completas eficientemente.

Árboles de clases de regresión

Con el fin de aumentar la flexibilidad del proceso de adaptación es posible determinar un conjunto apropiado de clases principales dependiendo de la cantidad de datos de adaptación que tengamos. Si sólo disponemos de una pequeña cantidad de datos, se generaría entonces únicamente una transformación global. Ésta se aplica a todas las Gaussianas que componen el modelo. Sin embargo, según aumenta la cantidad de datos se puede mejorar la adaptación incrementando el número de transformaciones a realizar. Cada una de estas transformaciones es más específica y se aplica a un determinado agrupamiento de Gaussianas. Por ejemplo, las Gaussianas se podrían agrupar según las clases de fonemas: silencio, vocales, nasales, fricativas, etc. Los datos de adaptación se utilizarían ahora para construir transformaciones más específicas para aplicarlas a esos grupos.

MLLR hace uso de un árbol de clases de regresión para agrupar las Gaussianas en el modelo, de manera que el conjunto de transformaciones a estimar se puede elegir de manera dinámica de acuerdo con la cantidad de datos de adaptación disponibles. Este árbol se construye para agrupar componentes que se encuentran próximas entre sí en el espacio acústico y se construye partiendo del modelo original independiente de locutor.

Los nodos terminales del árbol especifican las agrupaciones finales y se les denomina clases de regresión principales. Cada Gaussiana presente en el modelo pertenece a una de estas clases. La figura muestra un ejemplo de arbol de regresión con 4 clases terminales $\{C_4,C_5,C_6,C_7\}$.

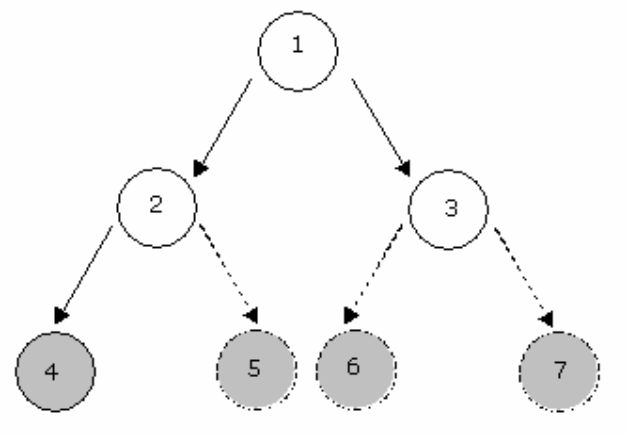

Figura 13. Ejemplo de árbol de regresión binario

Las flechas y nodos continuos indican que hay suficientes datos para que se genere una matriz de transformación utilizando los datos asociados a esa clase, mientras que las flechas y nodos discontinuos significan que no hay suficientes datos.

Para determinar en que nodos se va a realizar la transformación, se recorre el árbol desde la raíz y se genera una transformación para aquellos nodos que cumplen que:

1. Tienen datos suficientes **y**
2. Son nodos terminales **o** tienen algún hijo sin datos suficientes.

En el ejemplo anterior se generan entonces transformaciones para los nodos 2, 3 y 4 que llamamos W_2, W_3, W_4. Por lo tanto, a las componentes Gaussianas de cada clase principal de regresión se le aplica las matrices de transformación (medias y varianzas) de la siguiente manera:

$$\begin{Bmatrix} W_2 & \rightarrow & \{C_5\} \\ W_3 & \rightarrow & \{C_6, C_7\} \\ W_4 & \rightarrow & \{C_4\} \end{Bmatrix}$$

Es importante destacar por último que, una adaptación global corresponde al caso en que se tiene un árbol sólo con el nodo raíz.

Adaptación de modelos MAP

La adaptación de modelos ocultos de Markov también puede hacerse, como en el caso de los GMMs, con adaptación MAP (Maximim A Posteriori). La adaptación MAP, como ya se comentó en el apartado dedicado a la adaptación a los locutores en GMMs, consiste en el uso de información a priori sobre la distribución de los parámetros del modelo. Esta información a priori que se suele usar para realizar la adaptación se obtiene de los parámetros del modelo independiente de locutor. La fórmula de actualización de las medias para el estado j y la mezcla m es:

$\hat{\mu}_{jm} = \frac{N_{jm}}{N_{jm} + \tau} \overline{\mu}_{jm} + \frac{\tau}{N_{jm} + \tau} \mu_{jm}$, donde τ es el peso de la información a priori, N la probabilidad de ocupación de los datos de adaptación, μ_{jm} la media del modelo independiente de locutor y $\overline{\mu}_{jm}$ la media de los datos de adaptación.

Como se puede ver por la fórmula, si la probabilidad de ocupación de una componente Gaussiana N_{jm} es pequeña, entonces la estimación de la media se mantendrá cercana a la media de la componente independiente de locutor.

Una desventaja de la adaptación MAP es que requiere más datos que la adaptación MLLR para ser efectiva. Esto se debe a que la adaptación MAP se realiza a nivel de las Gaussianas componentes del modelo. Sin embargo, cuando disponemos de cantidades mayores de datos de adaptación, la adaptación MAP empieza a funcionar mejor que MLLR, debido a esta actualización detallada de cada componente. De hecho, se podrían combinar ambos procesos para conseguir mejores resultados, si sustituimos las medias previamente actualizadas con MLLR en μ_{jm} y procedemos entonces a realizar la adaptación MAP.

Aplicación de HMMs en Reconocimiento de Locutor dependiente de Texto:

En este subapartado se aborda el problema de cómo, a partir de lo visto anteriormente en los 3 problemas básicos de los HMMs (evaluar la probabilidad, entrenar un modelo, decodificar la secuencia de estados), cómo se aplica la teoría y se construye en la práctica un sistema de verificador de locutor.

Lo primero es partir de un conjunto de modelos acústicos (HMMs) independientes del locutor. Las unidades que se modelan pueden ser bien palabras enteras o fonemas.

Para la fase de entrenamiento es necesario reunir locuciones de los distintos locutores. Como ya se verá en los experimentos, tanto en la fase de entrenamiento como en la de verificación existe un compromiso entre la duración de las locuciones de entrenamiento y verificación y la precisión del reconocedor. El número de sesiones de entrenamiento también es determinante en el funcionamiento del sistema.

Partiendo del modelo independiente de locutor (λ_I) y con la información que nos aportan las grabaciones de los locutores se actualizan los parámetros del modelo independiente de locutor generándose unos nuevos modelos acústicos dependientes de cada locutor (λ_D). Este proceso de entrenamiento se puede llevar a cabo a través del método de reestimación Baum-Welch o mediante adaptación MLLR o MAP.

Únicamente cuando se ha finalizado la fase de entrenamiento es posible pasar a la de verificación. En ésta, el objetivo es obtener una puntuación para determinar, en función de un umbral, si la persona es aceptada o rechazada por el sistema. Para ello se enfrenta la frase a reconocer al modelo dependiente (λ_D) e independiente (λ_I) de locutor. La puntuación final se obtiene como el cociente de las puntuaciones acústicas obtenidas por el mejor camino (Qbest) en el reconocimiento de voz,

$P(O, Q_{best}|\lambda_D) / P(O, Q_{best}|\lambda_I)$,

y se compara con el umbral para tomar la decisión.

Cuando se realiza reconocimiento de locutor dependiente de texto es posible imponer restricciones a la gramática, de manera que admita únicamente una determinada secuencia de dígitos, por ejemplo, que corresponderían con la contraseña del usuario. Sin embargo, luego habrá que comprobar que realmente se ha reconocido lo que se esperaba.

La Figura 14 describe todo el proceso.

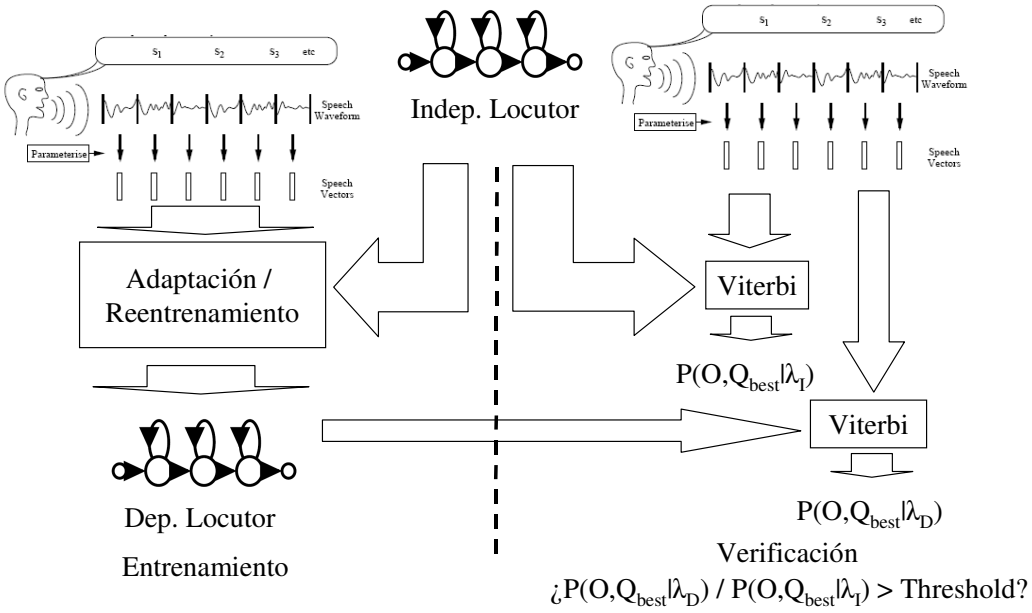

Figura 14. Aplicación de HMMs a un sistema de verificación de locutor

Capítulo 3

Diseño y Desarrollo

3 Diseño y Desarrollo

En este capítulo, dedicado al diseño y desarrollo, se va a describir los pasos seguidos para construir el reconocedor de locutor dependiente de texto, así como los medios que fueron necesarios.

3.1Medios disponibles

Este proyecto requiere una serie de medios que me fueron facilitados en su totalidad por el grupo ATVS para su realización. Estos medios necesarios se pueden resumir en bases de datos, software y máquinas.

3.1.1 Bases de datos

Las bases de datos son una parte esencial en el proyecto. A diferencia de lo que sucede en reconocimiento de locutor independiente de texto, campo en el que se organizan competiciones anuales promovidas por la organización NIST, no existe un equivalente en reconocimiento de locutor dependiente de texto. Esto, junto con el hecho de que muchas veces cada grupo de investigación utiliza sus propias bases de datos, dificulta la comparación de un sistema con otro. Sin embargo, con el fin de que se utilizara como marco de comparación entre sistemas surgió la base de datos YOHO.

YOHO

Surge con el objetivo de que sirva como marco de comparación entre sistemas de verificación de locutor y alentar la competición entre grupos de investigación. En un artículo [Campbell, 1995] se sugiere incluso un protocolo de pruebas a seguir.

La base de datos fue adquirida en un entorno de oficina, en condiciones muy controladas y bajo ruido. Se utilizo un auricular telefónico de alta calidad para capturar la voz. Está compuesta por 138 locutores, de los cuales 106 son hombres y 32 mujeres. El idioma de la base de datos es inglés americano, siendo la mayoría de locutores de la zona de Nueva York. Sin embargo, también se incluyen algunos locutores no nativos.

La base de datos se compone de datos de entrenamiento y de verificación. El entrenamiento se divide a su vez en 4 sesiones de 24 frases cada una. Los datos de verificación se adquirieron en 10 sesiones espaciadas una media de 3 días entre ellas y con 4 frases por sesión. En un escenario dependiente de texto, las frases son conocidas por el sistema, es decir, al usuario se le pide que las diga. El tipo de frases que componen la base de datos. La sintaxis utilizada en la base de datos es únicamente secuencias de tres pares de dígitos, a modo de contraseña de un candado de combinación. Un ejemplo de frase podría ser: “twenty-six, eighty-one, fifty-seven”.

TIMIT

La base de datos TIMIT fue diseñada para proporcionar habla acústico-fonética para el desarrollo y evaluación de sistemas de reconocimiento de habla automáticos. Se trata de habla capturada a través de un micrófono. La base de datos consta de 630 locutores que representan a los dialectos más importantes de inglés americano. De los 630 locutores, el 70% son hombres (438) y el 30% restante son mujeres(192). Cada locutor lee un total de 10 frases fonéticamente balanceadas, divididas en entrenamiento y test. Para el test se reservan 24 locutores, 2 hombres y una mujer de cada región dialectal, en el conjunto básico de test y 168 locutores en el conjunto extendido de test.

BIOSEC-Baseline

Se trata de una base de datos biométrica multimodal compuesta por 200 personas. Los rasgos biométricos que componen la base de datos son: huellas dactilares capturadas con 3 sensores diferentes, imágenes frontales de la cara, imágenes del iris y locuciones de voz grabadas tanto por un micrófono cercano como por el micrófono de una cámara web. Estos rasgos fueron adquiridos en 2 sesiones diferentes, separadas en el tiempo

entre una y cuatro semanas. Las condiciones ambientales, tales como iluminación o ruido de fondo, no fueron controladas para simular situaciones reales. Si nos centramos en las grabaciones de voz, cada locutor repitió cuatro veces un número específico de 8 dígitos que le fue asignado, así como 3 veces el número de otro usuario para simular situaciones en las que el impostor tiene acceso al número personal de un cliente. Los 8 dígitos se pronunciaron siempre dígito a dígito de manera continua y fluida. Estos fueron grabados tanto en español como en inglés. El número total de locuciones de voz presentes en la base de datos es por tanto: 2 sesiones x 200 usuarios x (4+3) x 2 sensores x 2 idiomas =11200.

ALBAYZIN

La base de datos ALBAYZIN fue concebida con el objetivo fundamental de fomentar el desarrollo de las tecnologías del habla en español. Está compuesta por 304 locutores, siendo la mitad hombres y la mitad mujeres. Todos ellos son hablantes de la variedad central del castellano y no presentan rasgos específicos de una zona geográfica o de un grupo social restringido. La base de datos se encuentra dividida en tres subcorpus: un corpus fonético, un corpus de aplicación formado por frases de una tarea de consulta a una base de datos geográfica y por último un corpus grabado en situaciones adversas. Vamos a utilizar el primero de ellos, que se basa en 200 frases fonéticamente balanceadas y que contiene un total de 6800 locuciones.

3.1.2 Software

El software del que disponíamos nos permitió y facilitó la realización del proyecto, por lo que también supuso un papel importante. Los paquetes de software que se utilizaron fueron:

- HTK: Es una herramienta para construir Modelos Ocultos de Markov. Los HMMs se pueden utilizar para modelar cualquier serie temporal y el núcleo es de propósito general. Sin embargo, HTK fue diseñado principalmente para crear herramientas para el procesado de la voz mediante HMMs, en particular reconocedores, por lo que la mayor parte de la infraestructura de HTK está dedicada a está tarea.

- SPHINX: Es un reconocedor de voz independiente de locutor y de amplio vocabulario. Además, está formado por una colección de herramientas de código abierto que permite construir sistemas de reconocimiento de voz.
- Software de evaluación de NIST: Programas facilitados por la organización NIST para el cálculo y representación de curvas DET en MATLAB.
- Núcleo GMMs ATVS: Conjunto de funciones y librerías desarrolladas por el grupo para crear reconocedores basados en GMMs.

3.1.3 Máquinas

Para la realización del proyecto fue necesario un ordenador funcionando con Linux (distribución Debian). Este ordenador tenía instalado todos los paquetes de software mencionados anteriormente. Se disponía además de una red interna que incluía a todos los ordenadores del grupo de trabajo, tanto los de uso personal como los de pruebas. De esta forma fue posible agilizar el proceso de entrenamiento y reconocimiento de modelos.

3.2 Diseño

Pasos seguidos para la construcción del reconocedor:

El diseño del reconocedor se puede resumir en los siguientes pasos:

1. Entrenamiento de los HMMs independientes de locutor: Reuniendo todas las locuciones presentes en la base de datos TIMIT, se entrena un modelo independiente de locutor en inglés. Se entrenaron modelos independientes de locutor cuya complejidad va en aumento, desde una Gaussiana por estado hasta 80 Gaussianas por estado. De manera similar, se utilizó la totalidad de la base de datos ALBAYZIN para entrenar los HMMs independientes de locutor en español.
2. Elaboración de un diccionario fonético con las palabras permitidas por el sistema.

3. Creación de la gramática de las frases: Por ejemplo, secuencias de 8 dígitos encadenados en la base de datos BIOSEC o secuencias de tres pares de dígitos en inglés encadenados con silencios intermedios opcionales en el caso de la base de datos YOHO.
4. Entrenamiento de los modelos dependientes de locutor: Primero es necesario seleccionar una serie de locuciones de cada locutor. Con éstas y partiendo del modelo independiente de locutor se entrenan los modelos dependientes de cada locutor.
5. Verificación de los locutores: Se reúnen frases de los distintos locutores a verificar y se comparan con los modelos de esos locutores y con el modelo independiente de locutor obteniendo una puntuación. En base a la puntuación obtenida se toma la decisión de aceptar o rechazar al locutor.
6. Normalización de las puntuaciones: Es posible aplicar distintas normalizaciones a las puntuaciones obtenidas.

Protocolo de Pruebas

Como no es posible hacer unas pruebas exhaustivas comparando cada modelo de locutor frente todas las locuciones de test de los distintos locutores, bien para el establecimiento de umbrales o para la evaluación del sistema, es necesario seleccionar minuciosamente un conjunto reducido de locuciones de test que nos den sin embargo una idea del funcionamiento del sistema. Para ello se elaboran los protocolos de pruebas para cada una de las bases de datos con las que vamos a trabajar.

Protocolo de pruebas para YOHO

- Se utilizan 6 archivos de entrenamiento (1 única sesión) para la prueba principal. (En otras pruebas se han utilizado, sin embargo, 24 y 96 archivos de entrenamiento, pertenecientes a 1 y a 4 sesiones respectivamente).
- 40 ficheros de test de cada locutor (todos)
- 1 fichero de test de cada uno de los locutores restantes (137) elegido aleatoriamente
- Por cada modelo del locutor = 40 + 137 =177 trials

- En total 138*177 trials= 24426 trials, de los cuales:
    - Enfrentamientos de usuario:138*40=5520
    - Enfrentamientos de impostor:138*137=18906

Protocolo de pruebas para BIOSEC

El corpus está formado por 200 locutores. Los 25 primeros y últimos corresponden al conjunto de desarrollo y los 150 restantes al conjunto de evaluación. Por lo tanto, se utilizan únicamente los 150 locutores pertenecientes al conjunto de evaluación.

- Se entrenan los modelos con una sola locución.
- Se entrenan 4 modelos por locutor.
- Enfrentamientos de test:
    - Usuarios:
        - Las 4 muestras de la primera sesión frente a las 4 muestras de la segunda sesión:
        - 150 locutores x 4 muestras x 4 muestras = 2400 comparaciones
    - Impostores:
        - La primera muestra de la primera sesión frente a la misma muestra del resto de usuarios, evitando comparaciones simétricas:
        - 150 locutores x 1 muestra x 149/2 locutores = 11175
    - En total: 11175 + 2400 = 13575 enfrentamientos.

Técnicas a comparar

Los experimentos que hemos realizado y que se presentan en el siguiente capítulo, tenían como objetivo observar el efecto que supone en los resultados el emplear distintas técnicas.

Una de las primeras pruebas que se realizaron fue comparar dos métodos de entrenamiento: la adaptación de modelos MLLR con la reestimación Baum-Welch. Otro experimento estudiaba la configuración óptima de los modelos, variando el número de Gaussianas por estado así como la cantidad de datos con los que fueron entrenados. Además, analizamos la influencia del número de clases de regresión al realizar la adaptación MLLR y el número de pasadas de reestimación en el método de reestimación Baum-Welch. Por último, se aplicaron distintas normalizaciones a los resultados.

Capítulo 4

Experimentos realizados

4 Experimentos realizados

Esta sección está dividida en dos apartados. El primero de ellos y más extenso, lo componen los experimentos realizados en la base de datos YOHO (inglés americano). El segundo apartado está formado por los experimentos llevados a cabo sobre la base de datos multimodal Biosec (español).

4.1 Experimentos sobre la base de datos YOHO

En ese apartado se presentan los experimentos realizados sobre la base de datos YOHO, que, por el tema de los mismos, se pueden dividir en tres secciones a su vez: comparar la reestimación Baum-Welch con la adaptación MLLR, aplicar una normalización a las puntuaciones y realizar fusión de dos sistemas.

4.1.1 Reestimación Baum-Welch versus Adaptación MLLR

4.1.1.1 Introducción

Esta primera sección de experimentos se realizó para verificar que se podían obtener mejores resultados adaptando los HMMs a las características de un determinado locutor mediante adaptación MLLR (Maximum Likelihood Linear Regression) que con la reestimación de los modelos fonéticos de cada locutor.

La reestimación Baum Welch, como ya se vio en el capítulo 2, se basa en el algoritmo de maximización de la expectación para la estimación de los parámetros de un HMM,

siendo éste el método más utilizado para entrenar un HMM y también en verifación de locutor dependiente de texto.

Se trata de un algoritmo iterativo donde en cada iteración se calcula la probabilidad de ocupación de todos los estados de un HMM para posteriormente actualizar sus parámetros. En su forma más general, se reestiman todos los parámetros de un HMM (probabilidades de transición, pesos de las Gaussianas, medias y varianzas), aunque es posible limitar la reestimación a sólo alguno de estos parámetros, como por ejemplo las medias. En ambos casos, el número de parámetros a estimar depende linealmente del número de Gaussianas del HMM.

Este hecho nos impone una limitación en la complejidad de los HMMs a utilizar, siendo habitual en reconocimiento de locutor dependiente de texto el uso de HMMs con un número de Gaussianas por estado que ronda entre una y cinco. Esto también repercute en la capacidad de los HMMs para modelar la acústica de la voz.

En los experimentos hemos utilizado HMMs de una a cinco Gaussianas por estado, llevando a cabo una y cuatro iteraciones en la reestimación en cada caso. También se han realizado experimentos en los que únicamente se han reestimado las medias.

Por otro lado, mediante la adaptación MLLR se realizan una serie de transformaciones para reducir las diferencias entre el modelo independiente del locutor inicial y las locuciones de cada locutor que se van a emplear en la adaptación. El efecto de estas transformaciones es desplazar las medias y modificar las varianzas del sistema inicial de manera que la probabilidad de que cada estado del sistema HMM inicial genere las locuciones de adaptación sea mayor.

La adaptación MLLR es una técnica especialmente desarrollada para la adaptación de HMMs independientes del locutor a la voz de un locutor en particular a partir de un número limitado de locuciones y por lo tanto se puede aplicar al problema de reconocimiento de locutor dependiente de texto. Además, en los casos en los que se cuenta con pocas locuciones para realizar la adaptación, la adaptación MLLR consigue mejores resultados si éstas se agrupan en clases y se transforman las medias de toda la clase utilizando para ello la misma transformación lineal. De esta manera, la adaptación MLLR reduce el número de parámetros a entrenar, pasando de depender linealmente con el número de Gaussianas a depender linealmente con el número de clases.

La ventaja adicional que posee la técnica MLLR es que es capaz de agrupar a su vez varias clases si no hay suficientes datos de adaptación. Esto, la convierte en una técnica

muy robusta de adaptación de HMMs a un locutor, incluso cuando se utilizan HMMs muy complejos.
Para los experimentos hemos realizado adaptación MLLR con 1 a 32 clases de regresión y HMMs fonéticos de 5 a 80 Gaussianas por estado.

4.1.1.2 Descripción de las pruebas y Resultados

Para comprobar cuál de los métodos obtiene mejores resultados nos hemos centrado en un escenario restrictivo en el que tanto para la adaptación MLLR o la reestimación Baum-Welch se utilizan únicamente 6 locuciones de la primera sesión de entrenamiento. Además, hemos comparado el rendimiento de ambos sistemas cuando se disponen de más locuciones de entrenamiento. Sin embargo, hemos realizado un mayor número de pruebas sobre el primer escenario ya que nos pareció que éste representaba más fielmente las condiciones que tendría un sistema real.

4.1.1.2.1 Reestimación Baum-Welch con pocos datos de entrenamiento

Los experimentos utilizando reestimación Baum-Welch se realizaron variando tres parámetros importantes: el número de Gaussianas por estado, el número de pasadas de reestimación y el número de locuciones de entrenamiento.
Se partió de modelos fonéticos independientes del locutor entrenados con la base de datos TIMIT, los cuales tenían desde una hasta ochenta Gaussianas por estado.
El entrenamiento se realizó con 6 locuciones, todas ellas de la primera sesión, realizándose tanto una como cuatro pasadas de reestimación.
Se realizó un alineamiento casi forzado, es decir, utilizando una gramática que sólo permitía una pronunciación pero que incluía silencios intermedios opcionales. Por ejemplo: (twenty six [silence] eighty one [silence] fifty seven).
Para la fase de verificación se utilizaron 40 ficheros de test de cada locutor y un fichero de cada uno de los restantes locutores (137) elegido de manera aleatoria. Se realizan, por lo tanto, para cada modelo de locutor: 40 + 137 = 177 enfrentamientos, siendo el número total de enfrentamientos 138*177 = 24426.

Con todo esto, los resultados obtenidos se resumen en la siguiente tabla.

		Gaussianas por estado				
		1	**2**	**3**	**4**	**5**
número de iteraciones	**1**	5.6	6.0	6.8	7.3	7.4
	4	6.4	7.9	10.0	14.4	16.6

Tabla 1. Resultados obtenidos sobre YOHO utilizando Reestimación Baum-Welch de HMMs independientes del locutor de 3 estados en función del número de Gaussianas por estado y el número de pasadas de reestimación.

La siguiente figura muestra las curvas DET para una pasada de reestimación.

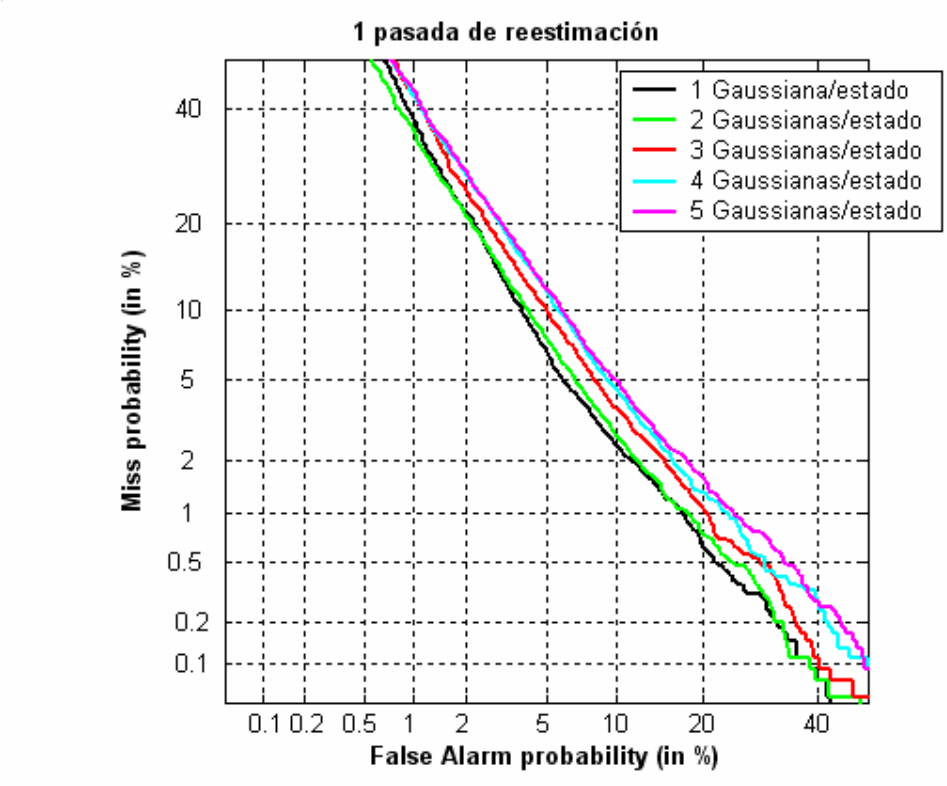

Figura 15. Curva DET con los resultados obtenidos sobre YOHO de realizar una pasada de reestimación Baum-Welch en función del número de Gaussianas por estado.

En la siguiente gráfica se pueden observar los resultados para 4 pasadas de reestimación.

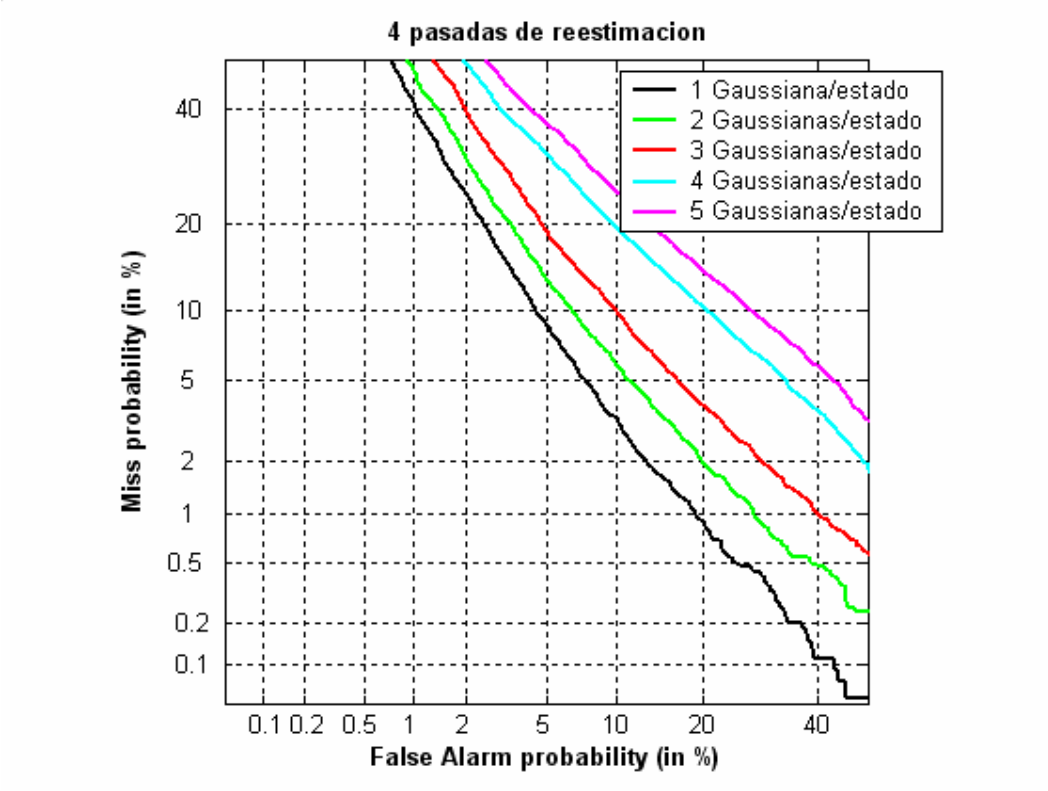

Figura 16. Curva DET con los resultados obtenidos sobre YOHO de aplicar 4 pasadas de reestimación Baum-Welch en función del número de Gaussianas por estado.

Se puede apreciar a la vista de las dos gráficas, que los resultados tienden a empeorar según aumenta el número de Gaussianas por estado, debido al limitado número de datos de entrenamiento y al incremento en los parámetros que es necesario estimar. Esto se puede ver claramente en el caso en que se realizan 4 pasadas de reestimación, en el que los resultados empeoran más rápidamente según aumenta el número de Gaussianas por estado. La razón es que si se realizan cuatro pasadas de reestimación con insuficientes datos de entrenamiento los modelos se degradan más.

Otro experimento consistía en ver el efecto que tenía el reestimar sólo las medias. Este experimento se realizó de manera similar a los anteriores, pero partiendo únicamente del modelo independiente de locutor de una Gaussiana por estado y se realizaron cuatro pasadas de reestimación. Sin embargo, el rendimiento que se obtuvo fue bastante

inferior, alcanzando un EER del 9.65%. A continuación se muestra la curva resultado obtenida.

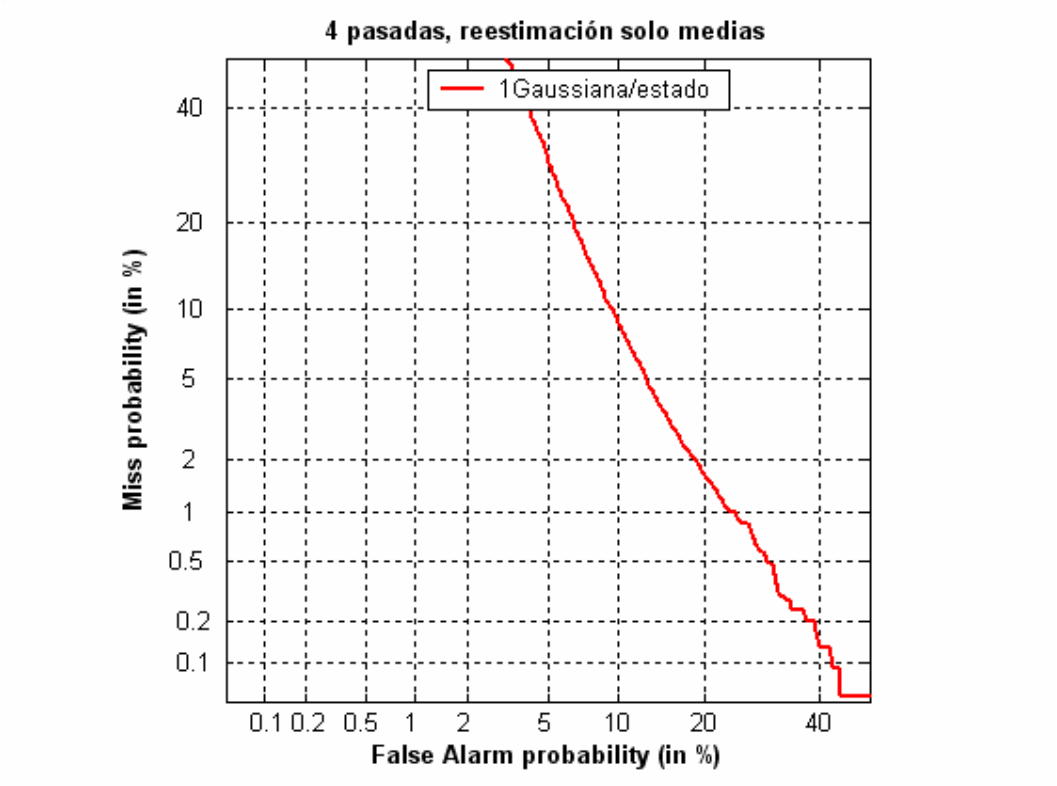

Figura 17. Curvas DET con los resultados obtenidos sobre YOHO al realizar 4 pasadas de reestimación Baum-Welch actualizando sólo las medias en modelos de una Gaussiana por estado.

4.1.1.2.2 Adaptación MLLR con pocos datos de entrenamiento

Una vez realizados los experimentos reestimando los HMMs con el algoritmo de Baum-Welch, se realizaron una serie de pruebas similares utilizando esta vez adaptación MLLR.

En la adaptación MLLR hemos variado el número de clases de regresión y el número de Gaussianas por estado. En estos primeros experimentos nos vamos a centrar, como en los anteriores, en el escenario más restrictivo de 6 locuciones de entrenamiento capturadas en una sola sesión.

El protocolo de pruebas es el mismo que el utilizado en reestimación Baum-Welch. Es decir, para un modelo de locutor dado, éste se va a enfrentar a las 40 locuciones de test del mismo locutor y a una locución de los restantes locutores elegida aleatoriamente.

El número de Gaussianas por estado tomó en los experimentos los valores 5, 10, 20, 40 y 80, mientras que al número de clases de regresión le asignamos los valores 1, 2, 4, 8, 16 y 32.

Fijando todos estos parámetros, los resultados de los experimentos en términos del EER se muestran en la siguiente tabla.

		Gaussianas por estado				
		5	**10**	**20**	**40**	**80**
clases de regresión	**1**	6.5	6.0	5.9	5.8	5.6
	2	5.3	4.8	4.7	4.6	4.3
	4	9.1	5.6	4.8	4.5	4.2
	8	9.1	5.4	5.1	4.6	4.2
	16	9.1	5.4	4.9	4.7	4.2
	32	9.1	5.4	4.9	4.7	4.2

Tabla 2. Resultados obtenidos sobre YOHO utilizando adaptación MLLR de HMMs independientes del locutor de 3 estados en función del número de Gaussianas por estado y el número de clases de regresión.

Observando los datos de la Tabla 2 y las gráficas que se muestran en el anexo (Figura 29 a Figura 34), se puede ver claramente que los resultados mejoran al aumentar el número de Gaussianas, al contrario que los resultados obtenidos con reestimación Baum-Welch, resumidos en la Tabla 1. Esta es precisamente una de las principales ventajas de utilizar adaptación MLLR en reconocimiento de locutor dependiente de texto, la posibilidad de emplear HMMs fonéticos más complejos y de mayor precisión.

Las gráficas anteriores nos confirman que manteniendo constante el número de clases de regresión, se produce una mejoría al incrementar el número de Gaussianas por estado. Si por el contrario, fijamos en cada gráfica el número de Gaussianas por estado y variamos el número de clases de regresión es posible analizar la influencia de las clases de regresión sobre los resultados (véase ANEXO, Figura 35 a Figura 39).

A la vista de las gráficas, se observa que el rendimiento tiende a mejorar cuando se pasa de una a dos clases de regresión y a partir de ahí se estabiliza con un número creciente de clases. Este comportamiento se puede deber al limitado número de datos de adaptación, que resulta insuficiente para entrenar más de dos transformaciones lineales. De hecho, a partir de dos clases, parece que la adaptación MLLR agrupa las distintas clases reduciendo el número efectivo de clases de regresión, con lo que el efecto que se consigue es casi el mismo que con sólo dos clases.

4.1.1.2.3 Adaptación MLLR y reestimación Baum-Welch variando el número de locuciones de entrenamiento

Para asegurarnos de la fiabilidad de los resultados obtenidos anteriormente, se repitieron las pruebas sobre escenarios menos restrictivos. En el primero de ellos se utilizaron 24 locuciones de entrenamiento de la primera sesión y en el segundo 24 locuciones capturadas en cuatro sesiones de entrenamiento distintas, haciendo un total de 96 locuciones de entrenamiento.

Con adaptación MLLR los resultados que se obtienen son similares a los del primer escenario: según aumenta el número de Gaussianas mejora el rendimiento.

En el caso de reestimación Baum-Welch, los experimentos demostraron que, al contrario que con 6 locuciones de entrenamiento, el rendimiento mejora según aumenta el número de Gaussianas por estado. [En el caso de 24 locuciones de entrenamiento y 4 Gaussianas por estado el EER es menor, sin embargo, para valores pequeños de falsa aceptación (zona de mayor interés) la curva de 5 Gaussianas por estado se haya por debajo de la de 4]. Esto se puede comprobar a la vista de la siguiente tabla resumen y de las gráficas con las curvas obtenidas.

Num. de locuciones entrenamiento (Num. de sesiones)	Gaussianas por estado				
	1	2	3	4	5
24 (1)	4.2	3.4	3.3	3.2	3.4
96 (4)	3.4	2.5	2.1	2.0	1.9

Tabla 3. Resultados sobre YOHO utilizando reestimación Baum- Welch en función de los datos de entrenamiento y del número de Gaussianas por estado.

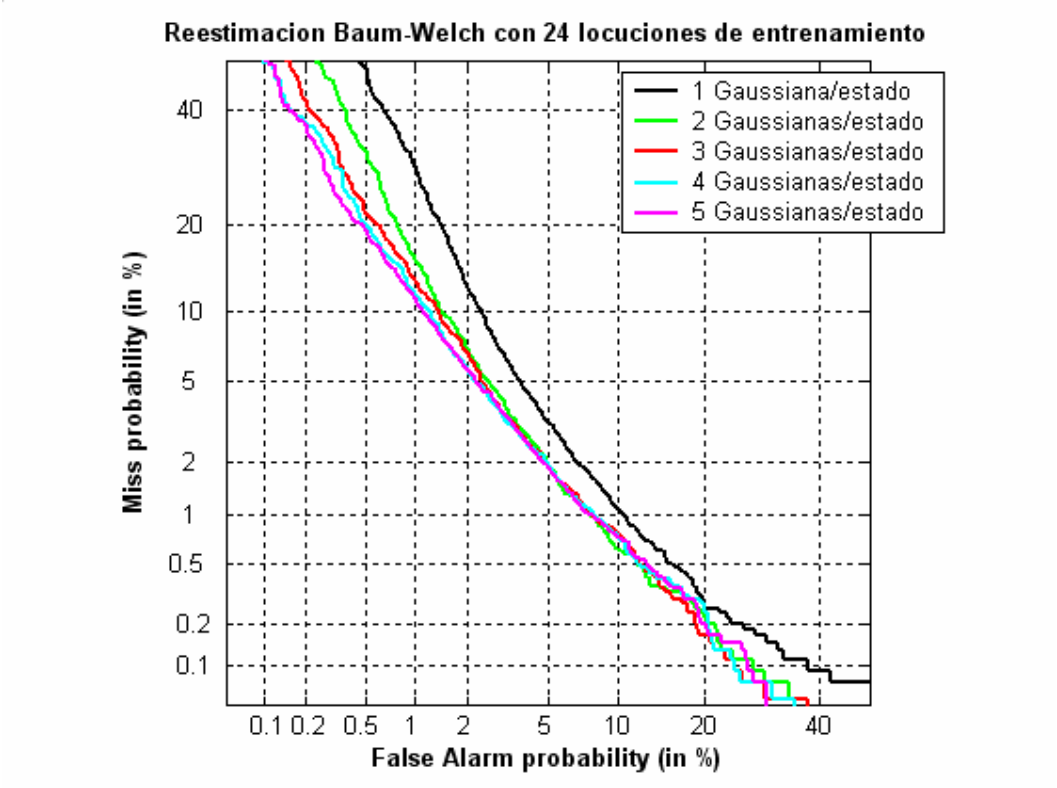

Figura 18.Curvas DET obtenida sobre YOHO tras realizar reestimación Baum-Welch con 24 locuciones de entrenamiento en función del número de Gaussianas por estado

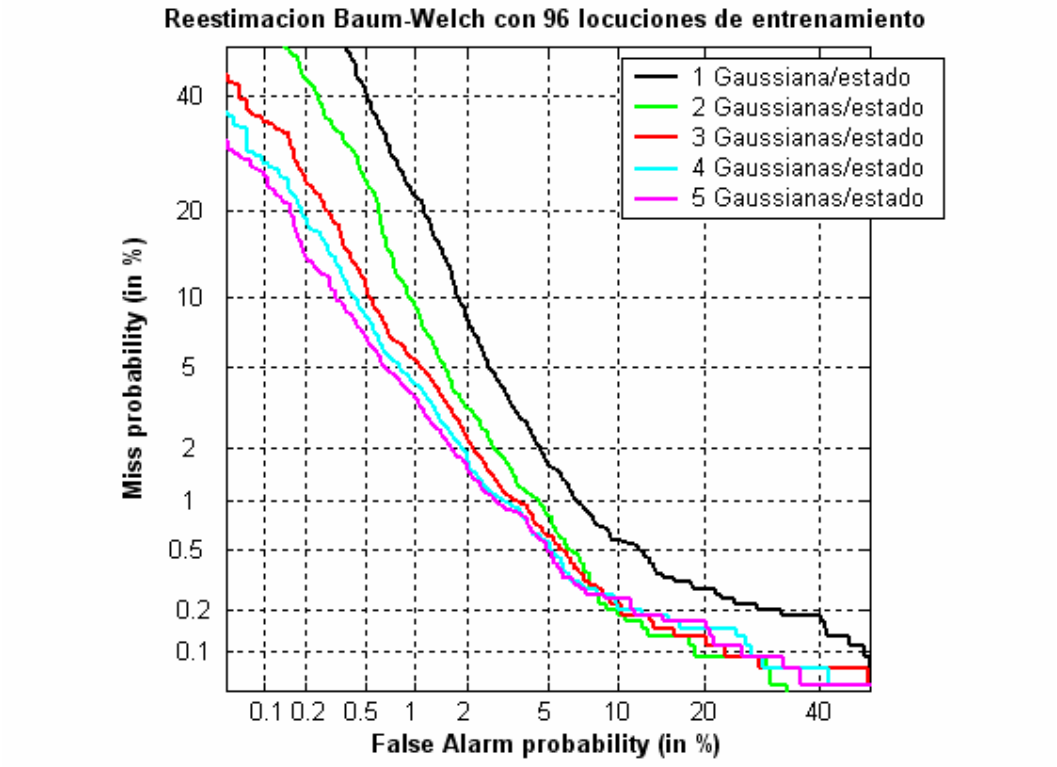

Figura 19. Curva DET obtenida sobre YOHO tras realizar reestimación Baum-Welch con 96 locuciones de entrenamiento en función del número de Gaussianas por estado

Seleccionando en cada caso la combinación (número de Gaussianas, pasadas de reestimación o clases de regresión) con mayor rendimiento, e incorporando los resultados que se obtuvieron para el primer escenario (6 locuciones de la primera sesión), se muestran de manera esquemática en la siguiente tabla y figura los resultados óptimos en función del número de locuciones de entrenamiento y de las sesiones a las que pertenecen.

Num. de locuciones entrenamiento (Num. de sesiones)	Adaptación MLLR	Reestimación Baum-Welch
6 (1)	4.6	5.6
24 (1)	2.1	3.2
96 (4)	0.9	1.9

Tabla 4. Resultados sobre YOHO utilizando reestimación Baum- Welch y adaptación MLLR en función de los datos de entrenamiento.

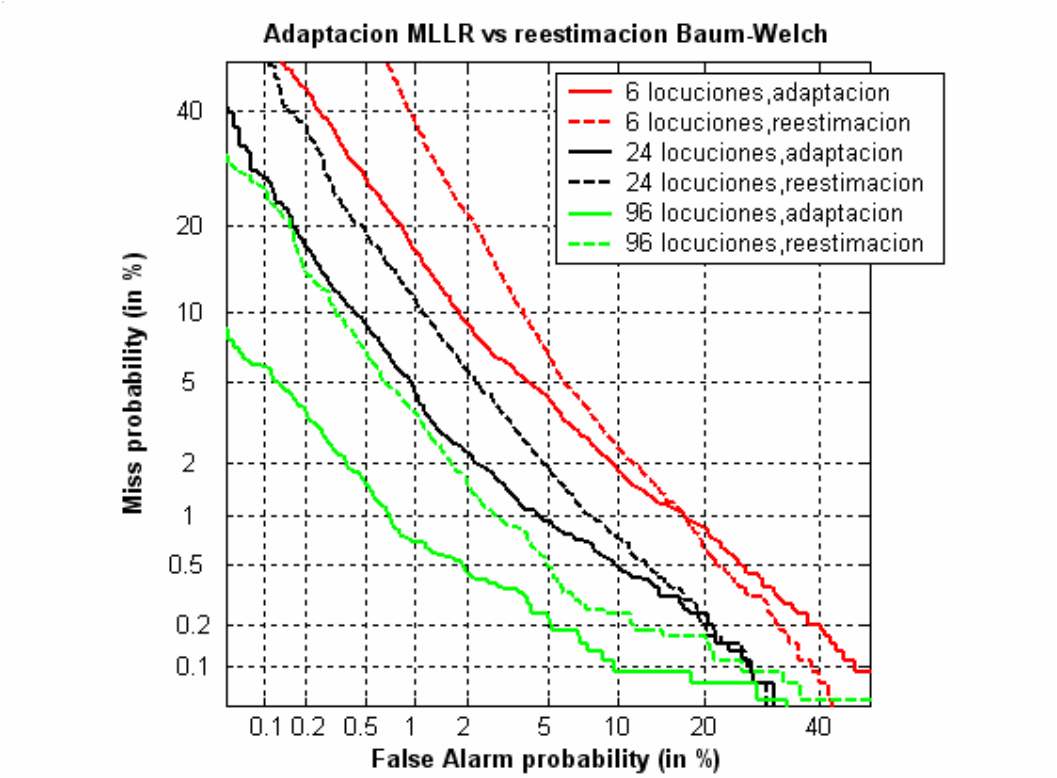

Figura 20. Curva DET obtenida sobre YOHO utilizando reestimación Baum- Welch y adaptación MLLR en función de los datos de entrenamiento.

4.1.1.3 Conclusiones

Comparando los resultados obtenidos con reestimación Baum-Welch y adaptación MLLR, Tabla 1 y Tabla 2 respectivamente, queda demostrado que se obtiene un mayor rendimiento utilizando adaptación MLLR para sistemas de reconocimiento de locutor dependiente de texto. Esto queda de manifiesto en especial cuando se emplean HMMs más complejos, es decir, con un mayor número de Gaussianas por estado. En estas circunstancias, la reestimación Baum-Welch no es capaz de competir con la adaptación MLLR, ya que para con tan sólo 6 locuciones de entrenamiento, según aumenta el número de Gaussianas por estado el rendimiento se degrada rápidamente. Por el contrario, MLLR puede trabajar con HMMs de un gran número de Gaussianas por estado, aún cuando sólo se utilizan pocas frases de entrenamiento y de esta manera aprovechar las ventajas que supone el uso de HMMs más complejos y detallados.

Cuando se utilizan más datos de entrenamiento, 24 y 96 locuciones, la conclusión anterior se sigue cumpliendo, esto es, que el método de adaptación MLLR funciona mejor que el de reestimación Baum-Welch. De hecho, según aumentan los datos de entrenamiento parece que las diferencias de rendimiento se incrementan, a favor de la adaptación MLLR como queda de manifiesto en la Figura 20.

4.1.2 Normalización de puntuaciones

4.1.2.1 Introducción

Una cuestión clave en la verificación automática de locutor realizada mediante métodos de modelado estadísticos como los GMMs o los HMMs es la normalización de las puntuaciones (score normalization), que cubre aspectos como la normalización de las verosimilitudes o normalización dependiente del tipo de auricular telefónico utilizado [Auckenthaler et al.,2000]. La normalización de las distribuciones de verosimilitudes de diferentes locutores se utiliza para encontrar umbrales globales independientes de locutor para el proceso de decisión mientras que la normalización dependiente del tipo de auricular trata de reducir los efectos de las diferentes condiciones ambientales.

El objeto de nuestro estudio va a ser la normalización de las verosimilitudes. Estas técnicas realizan una transformación de las puntuaciones de salida de un sistema de verificación de locutor con el objetivo de compensar las posibles diferencias que puedan existir en el rango de puntuaciones del conjunto de locutores debidas a variaciones en

las condiciones de cada uno de los enfrentamientos. Más concretamente TNorm o Test-Normalization es una técnica que trata de compensar la variabilidad en la verificación utilizando una cohorte fija de impostor.

El funcionamiento es el siguiente:

Supongamos que tenemos una secuencia de vectores de características $O = \{o_1, o_2, ..., o_N\}$ extraídos de una locución de test y un modelo de un locutor λt. Al comparar la locución de test con el modelo se obtiene una puntuación: $s(O, \lambda_t)$. Además esa misma locución de test se enfrenta a un conjunto de modelos de otros locutores, es decir, a un conjunto de impostores al que denominamos cohorte para obtener una serie de puntuaciones. A partir de estas puntuaciones de impostor se estiman la media y la varianza y se normaliza la puntuación inicial de la siguiente manera:

$$s_{TNORM}(O, \lambda_t) = \frac{s(O, \lambda_t) - \mu_{TNORM}}{\sigma_{TNORM}}$$

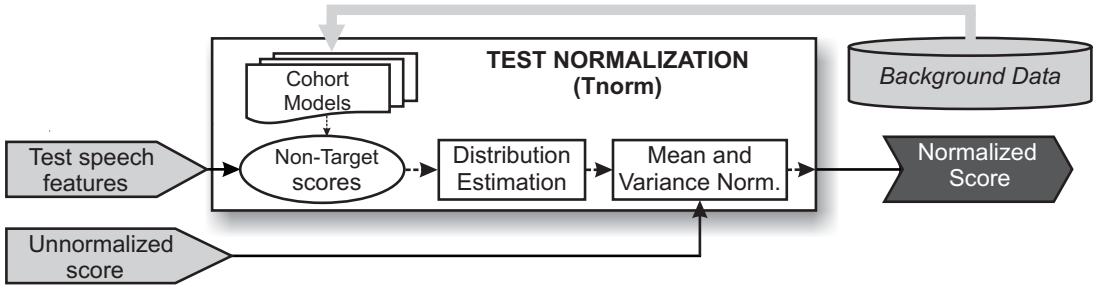

Figura 21. Diagrama de bloques de Tnorm, [Sturium et al., 2005]

4.1.2.2 Descripción de las pruebas y resultados

Dado que en la literatura existen muy pocas referencias al empleo de Tnorm en tareas de reconocimiento de locutor dependiente de texto, hemos realizado varias pruebas distintas hasta encontrar el modo de llegar a resultados satisfactorios.

Al contar con un número limitado de locutores, 138, el número de impostores de la cohorte no podía ser demasiado grande. Por lo tanto, fijamos la cohorte para los experimentos en 20 locutores, de los cuales 10 eran mujeres y 10 hombres.

Al igual que en experimentos anteriores, se entrenaron los modelos de los locutores con 6 locuciones de la primera sesión, partiendo de modelos fonéticos independientes del locutor de 40 Gaussianas por estado. El protocolo de pruebas para la fase de

verificación fue adaptado ligeramente para realizar la prueba. De manera similar a los experimentos anteriores, cada modelo de locutor se enfrentó con 40 locuciones de test legítimas y con una locución de test de los restantes locutores elegida aleatoriamente. Puesto que ahora sólo contamos con 118 locutores, el número total de enfrentamientos pasa a ser:

118* (40 + 1*117)=18526, de los cuales 4720 son enfrentamientos de usuarios y 13806 son enfrentamientos de impostor. A esto hay que añadir los enfrentamientos necesarios para aplicar Tnorm, con lo que el número total de enfrentamientos calculado anteriormente, 18526, hay que multiplicarlo por el tamaño de la cohorte, 20, resultando en 370520 enfrentamientos adicionales, lo que supone una gran carga computacional.

Con todo esto realizamos la prueba aplicando Tnorm a las puntuaciones obtenidas en cada enfrentamiento. En la siguiente figura se muestran gráficamente los resultados obtenidos, además de aparecer resumidos la tabla a continuación.

	EER	**FR para FA=1%**
Experimento Base	4.82	16.28
Tnorm	5.01	17.45

Tabla 5. Resultados sobre YOHO antes y después de aplicar Tnorm.

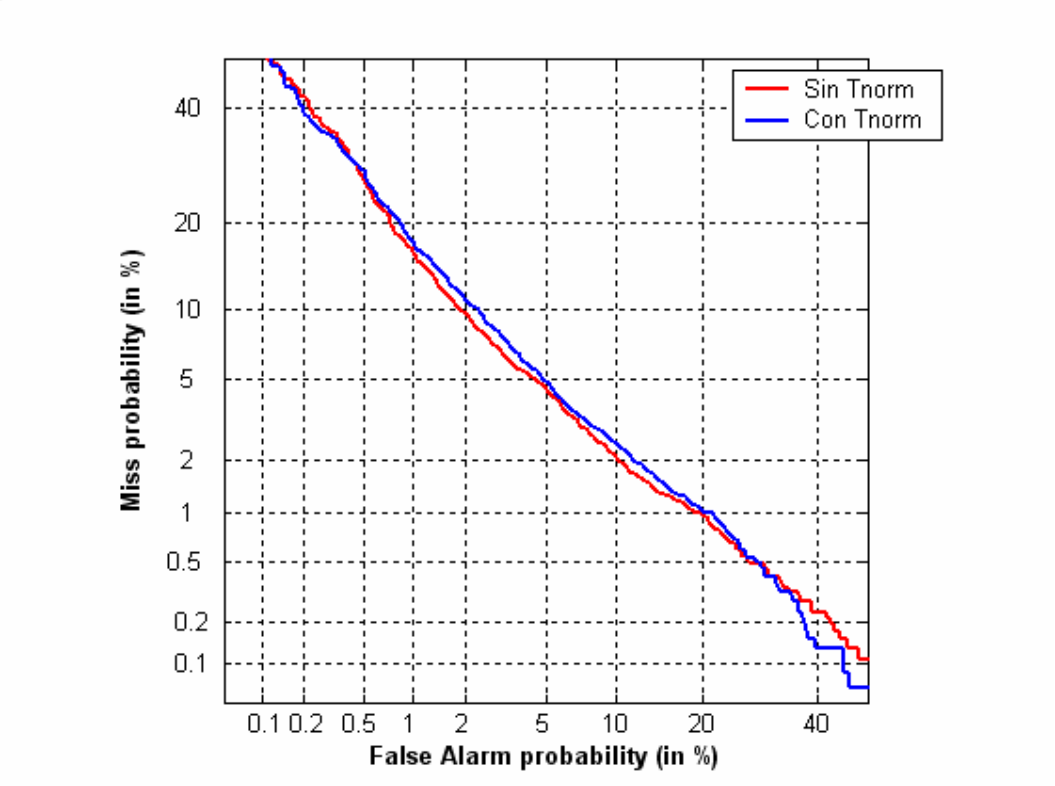

Figura 22.Curvas DET obtenidas sobre YOHO antes y después de aplicar Tnorm.

Como se puede apreciar en la figura, no se consigue ninguna mejora aplicando Tnorm de esta manera, sino que por el contrario el EER sufre un empeoramiento relativo del 3.9%. Esto se puede deber a que el tamaño de la cohorte no sea lo suficientemente grande como para estimar adecuadamente la varianza de la distribución. Esto concuerda con los resultados obtenidos de un experimento similar, en el que sin embargo, se entrenaban 4 modelos distintos de cada locutor de la cohorte. Con 4 modelos de locutor y 20 locutores formando la cohorte de impostores, se obtuvo un EER de 4.4%. Aquí se observa una mejora relativa en el EER del 7.9%.

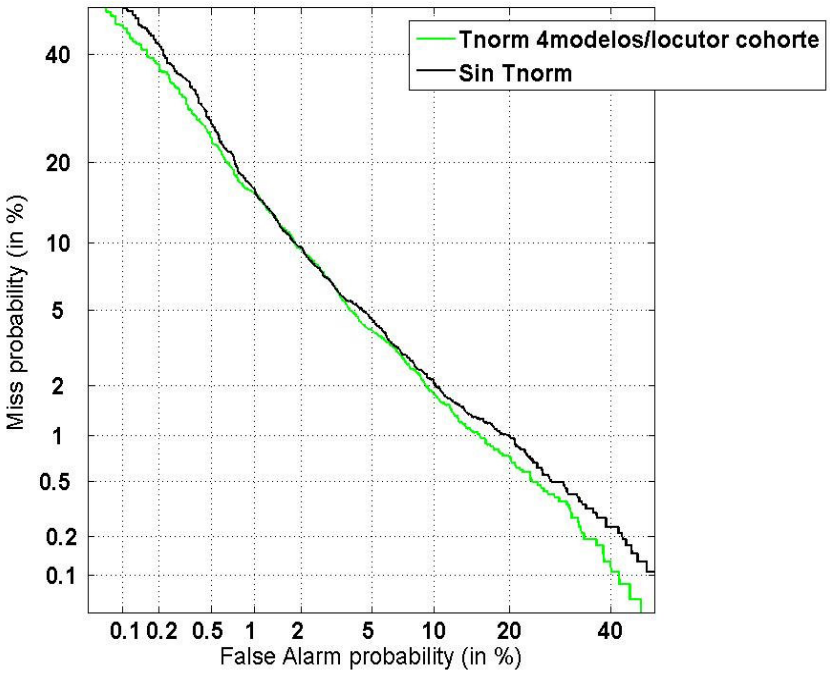

Figura 23.Curvas DET obtenidas sobre YOHO antes y después de aplicar Tnorm entrenando 4 modelos por cada locutor de la cohorte.

Por ello se deduce que a mayor número de modelos por locutor, el rendimiento del Tnorm va en aumento. Sin embargo, la mejora es muy lenta, comparada con la carga computacional adicional que esto supone. Por lo tanto, es clara la necesidad de encontrar un método distinto de aplicar Tnorm que obtenga resultados satisfactorios en reconocimiento de locutor dependiente de texto. El novedoso método que nosotros proponemos es el que explicamos a continuación.

Como se ha visto, aplicar Tnorm a las puntuaciones finales a la salida del detector no lleva a unos resultados satisfactorios debido a la gran variación existente dentro de cada uno de los ficheros de test, cuya puntuación total es la suma ponderada de la puntuación de cada uno de los fonemas presentes en la locución de test. Es por esto que los dos siguientes experimentos que se van a describir se centran en aplicar Tnorm a unidades más pequeñas, primero por fonemas y después, a un nivel aún más bajo, por estados.

La prueba a realizar es esencialmente la misma que en el experimento descrito al principio de este apartado. Se selecciona una cohorte de 20 locutores y se entrena un único modelo por locutor de la cohorte con 6 locuciones. El conjunto de usuarios sigue estando formado por 118 locutores, cuyos modelos han sido entrenados también con 6 locuciones. Los enfrentamientos para la verificación siguen el mismo protocolo

experimental que anteriormente. La diferencia reside en el momento de realizar la normalización para calcular la puntuación final. Para ello es necesario desglosar las puntuaciones obtenidas de cada enfrentamiento en fonemas. Entonces para cada fonema generado por el locutor a verificar, se calcula la media y varianza de las puntuaciones obtenidas en ese mismo fonema por la cohorte de impostores. Con estos datos, se normaliza la puntuación para ese fonema según la fórmula descrita más arriba. Por último, se calcula el promedio de las puntuaciones de cada fonema presente en el fichero y se obtiene la puntuación final.

La siguiente figura y la tabla que le acompañan recogen los resultados del experimento.

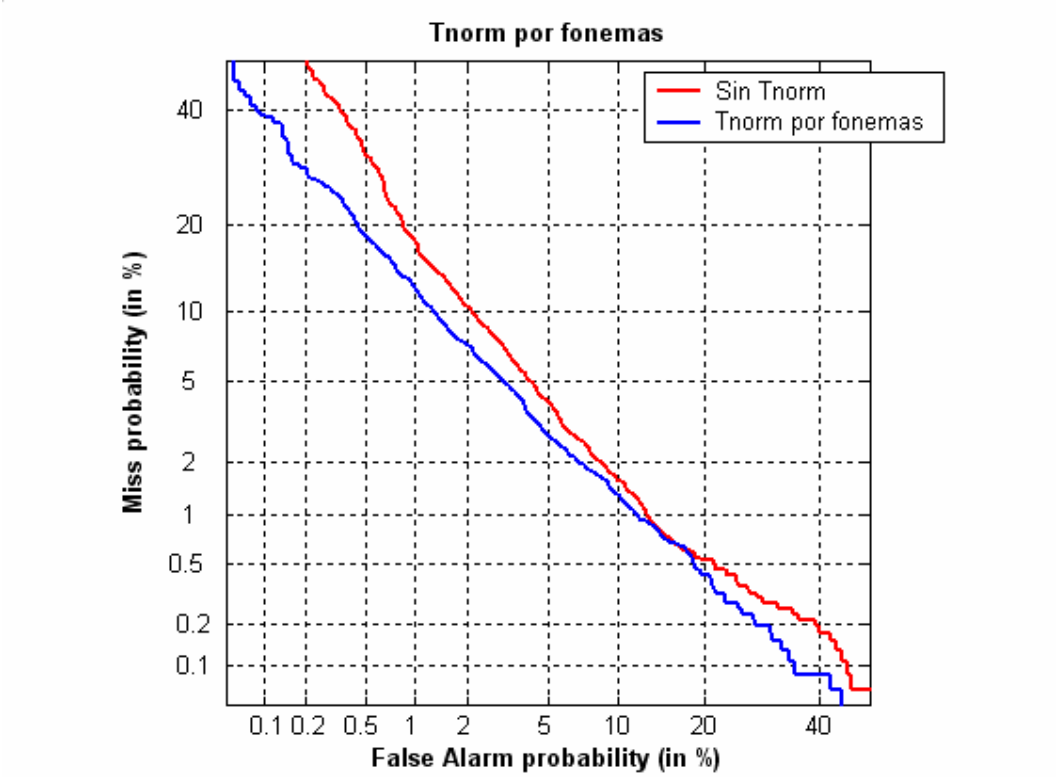

Figura 24.Curvas DET obtenidas sobre YOHO antes y después de aplicar Tnorm por fonemas.

	EER	FR para FA=1%
Experimento Base	4.51	17.81
Tnorm por fonemas	3.91	12.17

Tabla 6.Resultados sobre YOHO antes y después de aplicar Tnorm por fonemas.

En la gráfica se observa una inclinación de la curva a la que se le ha aplicado Tnorm en sentido contrario de las agujas del reloj a favor de una probabilidad de falso rechazo

menor para valores pequeños de falsa aceptación. Además la curva a la que se ha aplicado Tnorm se vuelve más recta, más cercana a una distribución Gaussiana. Estos dos efectos son característicos de Tnorm [Auckentaler et al., 2000].
Analizando los datos de la tabla vemos que éstos resultados reflejan una mejora relativa del 13.3 % en términos del EER. Además, si nos centramos en la zona de la curva de mayor interés, donde el valor de falsa aceptación toma valores pequeños, comprobamos que para un valor de falsa aceptación del 1% se consigue una mejora relativa del 31,67%.

De manera similar, también aplicamos Tnorm a unidades aún más pequeñas, a cada uno de los 3 estados que componen el modelo oculto de Markov para cada fonema que es pronunciado en cada fichero de test, con el fin de comprobar la validez de los resultados anteriores.

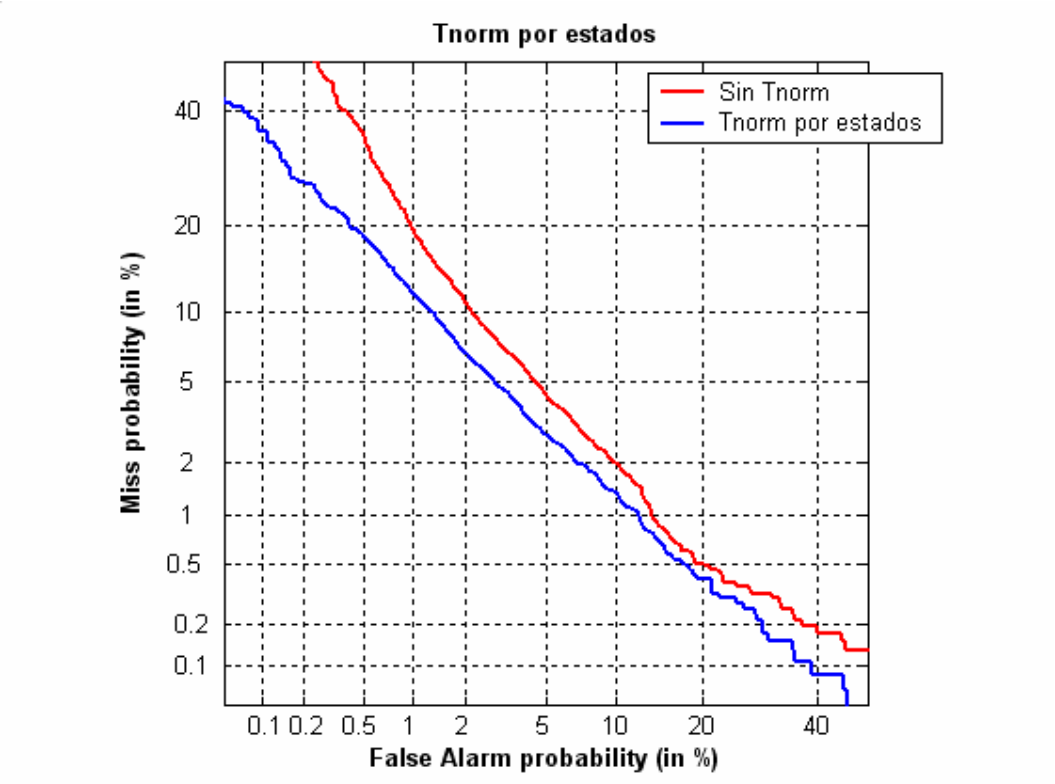

Figura 25.Curvas DET obtenidas sobre YOHO antes y después de aplicar Tnorm por estados.

	EER	FR para FA=1%
Experimento Base	4.80	19.63
Tnorm por estados	3.85	11.81

Tabla 7.Resultados sobre YOHO antes y después de aplicar Tnorm por estados.

La figura anterior muestra la misma tendencia que observábamos al aplicar Tnorm por fonemas. La curva se inclina en sentido contrario a las agujas del reloj y se vuelve más lineal. Se incrementa ligeramente la mejora con respecto al Tnorm por fonemas. En este caso, la mejora en términos del EER con respecto al experimento sin Tnorm alcanza el 19.8%. El valor de falso rechazo para una falsa aceptación del 1% se experimenta una reducción relativa del 39.84 %.

4.1.2.3 Conclusiones

En esta sección hemos visto la necesidad de modificar la manera de aplicar Tnorm con respecto a la manera habitual empleada en reconocimiento de locutor independiente de texto para adaptarlo a tareas dependientes de texto. Esta cuestión está poco documentada hasta la fecha, centrándose la mayoría de los estudios en tareas independientes de texto. En [M.Hebert et al.,2005] se describe un método de normalización basado en Tnorm que introduce un término correctivo para compensar los casos en los que los modelos de la cohorte no fueron entrenados con las mismas locuciones con las que se va a verificar. Sin embargo, el método que hemos empleado en nuestros experimentos tiene la ventaja de ser invariable frente a diferencias en las locuciones de entrenamiento y de test de los modelos de la cohorte. Esto se debe a que la normalización se realiza a nivel de fonemas, con lo que las locuciones con las que se entrenaron los modelos no necesitan ser iguales a las que se utilizan en verificación. La única restricción con respecto a las frases de entrenamiento es que éstas deberían ser fonéticamente balanceadas.

En conclusión, el método presentado aquí es una manera sencilla y eficaz de resolver el problema de aplicar Tnorm en reconocimiento de locutor dependiente de texto.

4.1.3 Fusión HMM/GMM

4.1.3.1 Introducción

La fusión de dos sistemas permite combinar las ventajas de estos dos sistemas para realizar un mejor reconocimiento, al disponer de más información de los locutores. De esta manera es posible combinar distintos rasgos biométricos, mediante la fusión de las puntuaciones obtenidas por los sistemas individualmente sobre cada uno de los rasgos. La fusión se realiza normalmente mediante reglas sencillas de combinación, como la suma o el producto [Kittler et al., 1998], o utilizando clasificadores entrenados [Verlinde et al., 2000], tales como Redes Neuronales o Máquinas de Vectores Soporte.

Para realizar la fusión suma, que es la que vamos a utilizar en los experimentos, es necesario primero normalizar las puntuaciones de cada uno de los dos sistemas de manera que se encuentren en el mismo rango y se puedan sumar. Para ello se calcula la media y varianza de la distribución de los impostores y se normalizan las puntuaciones tanto de usuarios como de impostores con la siguiente fórmula:

$$s = \frac{s - \mu_i}{\sigma_i}$$

Esto no es más que aplicar Tnorm, cuyo resultado es proyectar la distribución de puntuaciones de los impostores sobre una distribución Gaussiana de media nula y varianza unidad.

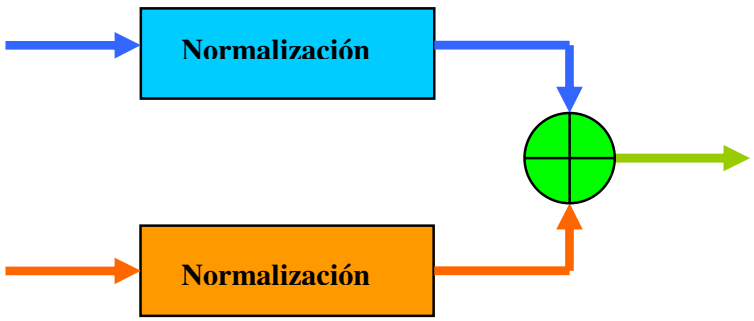

Figura 26. Diagrama de bloques que muestra el proceso de fusión.

4.1.3.2 Descripción de la prueba y Resultados

Tanto el sistema HMM como el sistema GMM fueron entrenados con 6 locuciones de entrenamiento de la primera sesión. El protocolo de pruebas que seguimos fue el mismo que en el resto de experimentos.

Descripción del sistema HMM:

Se utilizaron HMMs fonéticos adaptados al locutor mediante adaptación MLLR, a partir de un HMM independiente de locutor de 40 Gaussianas por estado entrenado con la base de datos TIMIT.

Descripción del sistema GMM:

Se utilizó la siguiente parametrización:

- Enventanado Hamming de 20 ms, solapadas 10 ms entre sí.
- 20 filtros de magnitud según la escala mel (0-4000 kHz).
- 38 coeficientes por segmento de voz (19 MFCC + delta).
- Filtro paso banda entre 300 y 3300 Hz.
- Filtro CMN y Rasta.
- Se aplicó un detector de actividad vocal estático basado en energía para eliminar los silencios.

El UBM se entrenó con toda la base de datos YOHO (entrenamiento + verificación) usando 1024 Gaussianas y estimación ML (maximum likelihood) mediante el algoritmo EM (expectation-maximization). Esto hace que los resultados obtenidos con los GMM sean un poco "optimistas" ya que se han empleado en el entrenamiento los mismos datos que se van a emplear en la evaluación y existe una coincidencia perfecta en las condiciones acústicas de la voz de entrenamiento y evaluación. Los modelos de locutor consistieron en GMMs de 1024 mezclas adaptados (sólo medias) a partir del UBM mediante la técnica MAP (Maximum A Posteriori) con una única iteración. Para la adaptación se emplearon 6 locuciones de la primera sesión por locutor. Para la identificación se emplearon únicamente las 5 Gaussianas más pesadas por trama para el cálculo de verosimilitudes. No se realizó ninguna normalización de scores.

Una vez realizadas las pruebas con cada uno de los sistemas de manera individual y sacadas las puntuaciones, se procede a la normalización según se explicó anteriormente y a la suma de ambas puntuaciones. Con todo esto, la curva DET que se obtiene es la que se muestra en la siguiente figura.

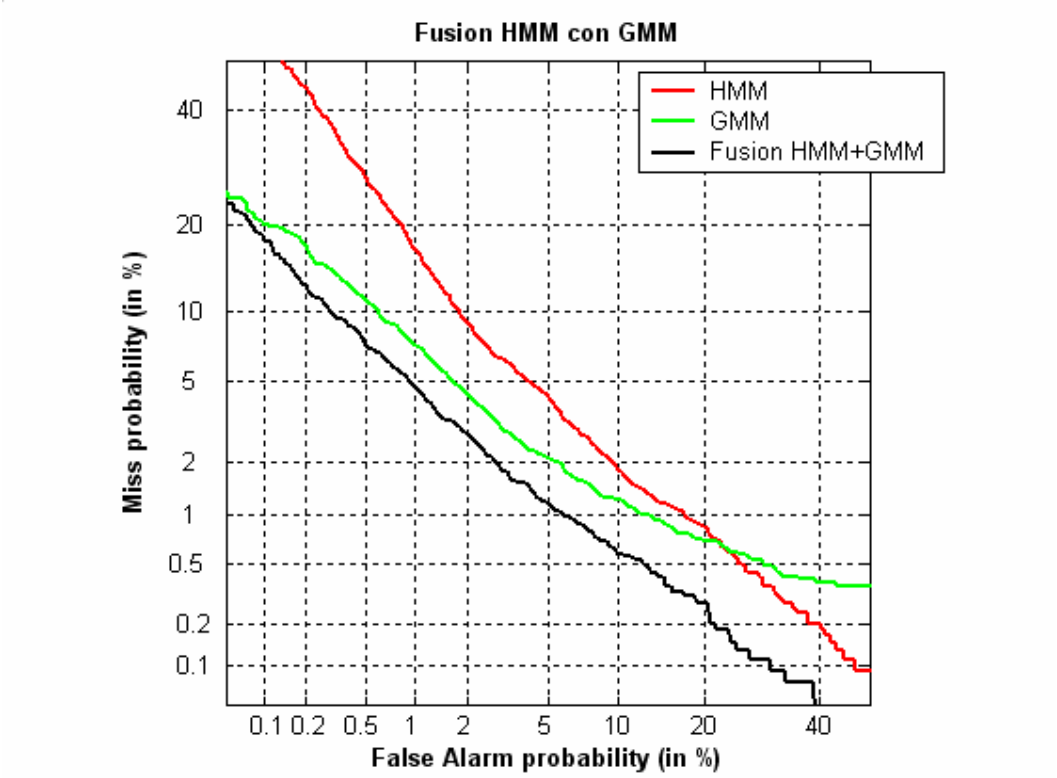

Figura 27. Curvas DET obtenidas sobre YOHO antes y después de realizar la fusión de los sistemas individuales basados en GMMs y HMMs.

Sistema	EER
HMM	4.63
GMM	3.03
Fusión	2.42

Tabla 8. Resultados obtenidos sobre YOHO antes y después de realizar la fusión de los sistemas individuales basados en GMMs y HMMs.

4.1.3.3 Conclusiones

Como era de esperar, al fusionar ambos sistemas se produce una mejora significativa en los resultados (véase

Tabla 8). Esto es debido, como ya se apuntó antes, a que la fusión incorpora más información sobre la distribución de usuarios e impostores y se puede beneficiar de las ventajas de los dos sistemas, obteniendo como resultado un mayor rendimiento en la verificación de los locutores. La comparación de los sistemas GMM y HMM es claramente favorable al GMM. Este resultado hay que tomarlo con cautela, porque como comentábamos anteriormente el GMM empleado se ha entrenado con los datos de

la evaluación y por tanto los resultado son algo "optimistas". Además, el GMM no permite comprobar que el texto dicho por el usuario sea el esperado (su palabra clave o la frase solicitada) mientras que los HMMs permiten comprobarlo al mismo tiempo que se comprueba si la voz es la del usuario registrado.

4.1.4 Comparación de nuestro sistema frente a otros

En [Campbell, 1995] se muestra una tabla con la comparación de los resultados de distintos sistemas sobre la base de datos YOHO. Resulta difícil comparar unos sistemas con otros, puesto que cada uno utiliza protocolos de prueba distintos. Sin embargo, nuestros resultados con YOHO utilizando todos los datos de entrenamiento nos sitúan cercanos a estos resultados. En nuestras pruebas utilizando todo el material de entrenamiento (96 frases por usuario), como hacen la mayoría de los experimentos recogidos en esta tabla, hemos obtenido un EER del 0.9 %. Este resultado se ha obtenido sin realizar ningún tipo de normalización de puntuaciones. Hemos comprobado que TNorm consigue mejorar los resultados, por lo que aplicándolo podríamos reducir nuestro resultado del 0.9%. No lo hemos hecho porque estábamos más interesados en una situación más real en la que se dispone de pocos datos de entrenamiento. También podríamos mejorar nuestro sistema combinando el sistema GMM con el resultado del 0.9% de EER obtenido con HMMs, pero nuevamente no se ha realizado esta prueba por tratarse de una condición poco realista.

	Verification EER	Speaker Id closed-set
ITT's CSR	1.7%	
ITT's NN	0.5%	
MIT/LL's	0.51%	0.8% error
GMM	0.2%m, 1.8%f	1.1 avg rank
Rutgers' NTN	0.65%	
Rutgers' HMM		1.36% error
		1.05 avg rank
Rutgers' LVQ		0.36% error
		1.03 avg rank
Campbell's		1.15% error
divergence		1.01 avg rank

Tabla 9.Comparación del rendimiento de distintos sistemas sobre YOHO.

4.2 Experimentos sobre la base de datos BIOSEC

Esta última sección muestra los experimentos que hemos realizado sobre la base de datos BIOSEC, con el fin de ver si los resultados obtenidos sobre YOHO seguían siendo consistentes sobre BIOSEC. Sin embargo, el protocolo de pruebas es distinto debido a las diferencias entre las dos bases de datos.

4.2.1 Adaptación MLLR vs Reestimación Baum-Welch

4.2.1.1 Introducción

Al igual que hacíamos con la base de datos YOHO, el objetivo de este experimento es determinar cuál de los dos métodos de entrenamiento funciona mejor.

4.2.1.2 Descripción de la prueba y Resultados

Como esta prueba se realizó después de haber realizado los experimentos sobre YOHO, si hacemos uso de las conclusiones extraídas de los experimentos anteriores es posible realizar una versión simplificada y reducida de la misma. De esta manera, únicamente seleccionamos aquella configuración (Número de Gaussianas, Clase de regresión/ Pasadas de reestimación) que mejores resultados obtuvo en adaptación MLLR y reestimación Baum-Welch. Por lo tanto, para la adaptación Baum-Welch se partió de modelos independientes de locutor de español de 40 Gaussianas por estado y se aplicaron las matrices de transformación a 2 clases de regresión. Por otro lado, la reestimación Baum-Welch se realizó en una sola pasada a partir de modelos independientes de locutor en español de una Gaussiana por estado.

Como ya se vio en el apartado 3.2, se entrenan 4 modelos por cada locutor, utilizando para ello una única locución. Para estos experimentos hemos seleccionado las locuciones en español que fueron grabadas por un micrófono de habla cercana.

Los enfrentamientos de test se realizaron siguiendo el protocolo de pruebas descrito anteriormente. Es decir, para los enfrentamientos de usuario se comparan las 4 locuciones de la segunda sesión con los 4 modelos de locutor entrenados con la primera sesión. Además, se enfrenta una locución del resto de usuarios al primer modelo del locutor, obteniendo así los enfrentamientos de impostor.

Con todo esto, los resultados que obtuvimos se resumen en la siguiente tabla:

Reestimación Baum-Welch	Adaptación MLLR
8.17 %	1.68 %

Tabla 10. Resultados sobre BIOSEC para reestimación Baum-Welch y adaptación MLLR para locuciones en español y adquiridas con un micrófono cercano.

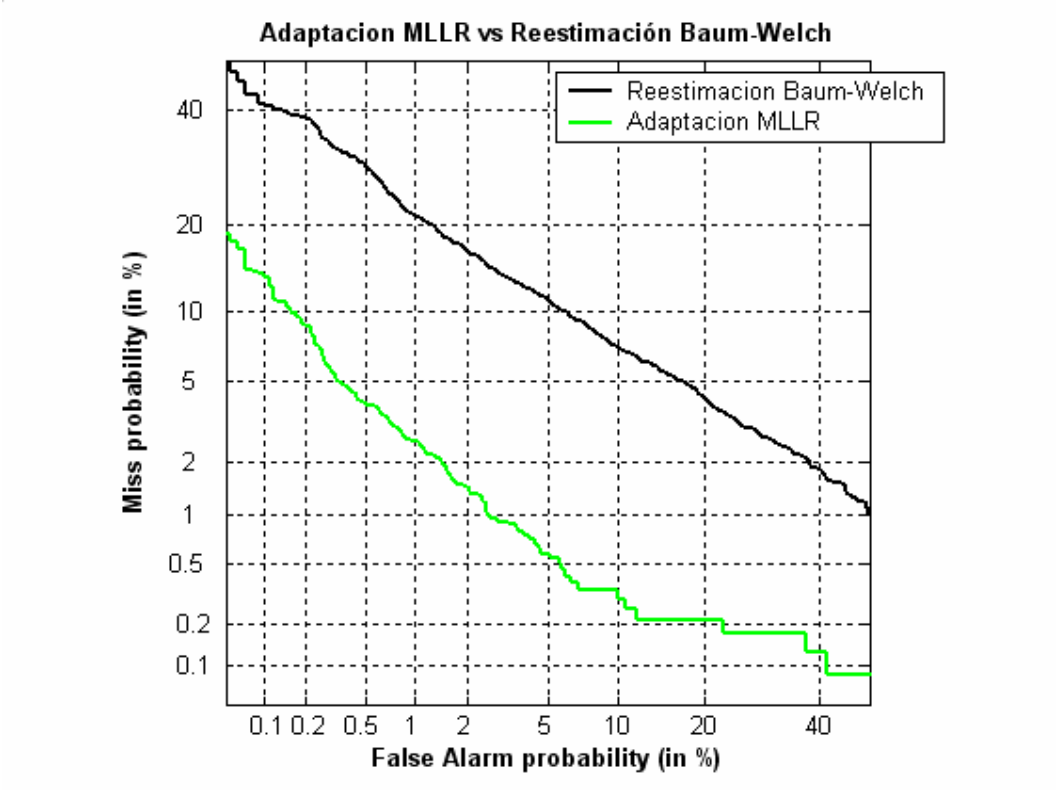

Figura 28.Resultados sobre BIOSEC para reestimación Baum-Welch y adaptación MLLR para locuciones en español y adquiridas con un micrófono cercano.

4.2.1.3 Conclusiones

La tabla y la figura muestran una gran diferencia en el rendimiento de ambas técnicas a favor de la adaptación MLLR. Esto confirma los resultados que se obtuvieron sobre la base de datos YOHO, de que se producía una mejora en el entrenamiento si adaptábamos los modelos independientes de locutor a los distintos locutores mediante la técnica MLLR en vez de realizar pasadas de reestimación. En este caso, además, con la base de datos BIOSEC y el protocolo de pruebas establecido para ésta, la mejora que se obtiene es mucho más significativa (6.5 puntos en el EER). Esto se debe posiblemente a que sólo entrenamos con una frase y MLLR funciona bien con pocos datos de entrenamiento.

Capítulo 5

Conclusiones y Trabajo Futuro

En el presente proyecto se han estudiado los sistemas de reconocimiento de locutor dependiente de texto, en concreto, aquellos construidos en base a los modelos ocultos de Markov. Seleccionamos esta técnica de modelado frente a otras también habituales en reconocimiento de locutor dependiente de texto, como puede ser Dynamic Time Warping, por ajustarse perfectamente a las características inherentes del reconocimiento dependiente de texto; éstos permiten el desarrollo de soluciones de texto variable de una manera sencilla, con sólo utilizar HMMs para modelar cada uno de los fonemas.

Sin embargo, se trata de una técnica compleja computacionalmente pero a la vez muy flexible, lo que supone que disponemos de muchos grados de libertad. Esto ha dado lugar a numerosos experimentos para llegar a construir de manera óptima un sistema de reconocimiento de locutor basado en HMMs.

Los primeros experimentos trataban de determinar qué método de entrenamiento conseguía mejores resultados. Para ello se comparó el funcionamiento de sistemas entrenados de forma clásica, mediante el método de reestimación de Baum-Welch, frente a sistemas en los que los modelos de locutor habían sido adaptados mediante MLLR. A raíz de estos experimentos, pudimos concluir que el sistema óptimo debía implementarse utilizando adaptación de locutores con MLLR y seguir estudiando la influencia de otros parámetros partiendo de esta base. Esto se debe, a que el método de adaptación MLLR, mediante el agrupamiento de componentes que se encuentran cercanas en el espacio acústico, le permite beneficiarse de las ventajas de emplear HMMs más complejos y detallados, sin que esto repercuta negativamente en el tiempo de máquina necesario. A pesar de que nuestro objetivo era un sistema que trabajase con

pocos datos de entrenamiento, repetimos la prueba aumentando la cantidad de datos y comprobamos que los resultados anteriores se seguían manteniendo.

Otro tema al que le hemos dedicado nuestra atención, ha sido el aplicar con éxito normalizaciones a los resultados finales, al igual que se realiza en las tareas de reconocimiento de locutor independiente de texto. La solución se encontró al normalizar las puntuaciones obtenidas en cada fonema o estado, obteniendo una mejora notable en los resultados finales.

Con todo esto, creemos que hemos cumplido con los objetivos que nos marcamos inicialmente, desarrollando un sistema de reconocimiento de locutor dependiente de texto y evaluarlo tanto con la base de datos YOHO, como con la base de datos BIOSEC, cuyo rendimiento se encuentra al nivel del estado del arte.

Como líneas de trabajo futuro, nos gustaría probar otro método de adaptación de modelos de locutor: adaptación MAP, o el utilizar MLLR y MAP en cadena, ya que en [The HTK Book, 2005] se auguran mejores resultados con esta combinación de ambas técnicas. Otro aspecto interesante para analizar sería el uso de modelos fonéticos dependientes del contexto.

Capítulo 6

Referencias

Auckenthaler, R., Carey , M., Lloyd-Thomas, H., "Score Normalization for Text-Independent Speaker Verification Systems", Academic Press, 2000.

Baum, L.E. and Petrie, T. " Statistical inference for probabilistic functions of finite state Markov chains" *Ann. Math. Stat.,* Vol. 37. pp 1554 – 1563, 1.966.

Baum, L.E. and Egon, J.A. , " An inequality with applications to statistical estimation for probabilistic functions of Markov process and to a model ecology ", *Bull. Amer. Meterol. Soc*., vol.73, pp 360 – 363, 1.967.

Baum, L.E. and Sell, G.R., " Growth functions for transformations on manifolds", *Pac. J. Math,* vol.27,No. 2 pp.211- 227, 1968.

Baum, L.E., Petrie, E, Soules, G. and Weiss, " A maximisation technique occurring in the statistical analysis of probabilistic functions of Markov Chains ", *Ann. Math. Stat.,* vol. 41, No. 1, pp164 – 171., 1970.

Bimbot, F., Hutter,H.P., Jaboulet, C., Koolwaaij,J., Linberg,J. and PIERROT, "speaker verification in the telephone network: research activities in the CAVE projet", in *Proc Eurospeech ,* 1997, pp 971 –974

Bimbot, F., "Speaker Verification Technology in the CAVE project", CAVE Project LE 1930 ,Work Package 4 : Deliverable 4, 1997.

BIOSEC, Deliverable 5: Extended Multimodal Database and Testing Protocol, 2005

Campbell, J. P. Jr., Speaker recognition, Biometrics: Personal Identification in Networked Society , Capítulo 8,Kluwer Academic Publishers, 1999.

Campbell, J.P. and Higgins, A., Yoho speaker verification (ldc94s16). http://www.ldc.upenn.edu.

Campbell, J.P., "Testing with the YOHO CD-ROM voice verification corpus" in *Proc. ICASSP 1995*, vol. 1, pp 341 –344

Colás Pasamontes, J., *Estrategias de incorporación de conocimiento sintáctico y semántico en sistemas de comprensión de habla continua en español*. Estudios de Lingüística del Español, Volumen 12, 2001.

6. Referencias

Fierrez, J., Ortega-Garcia,J., Torre Toledano, D., Gonzalez-Rodriguez, J.,"Biosec baseline corpus:A multimodal biometric database", *Pattern Recognition*, Vol. 40, n. 4, pp. 1389-1392, Abril 2007.

Frédéric Bimbot, "Speaker Verification Technology in the CAVE project", CAVE Project LE 1930 ,Work Package 4 : Deliverable 4, 1997.

Hébert , M., Boies, D., "T-Norm for Text-Dependent Speaker Verification Applications: Effect of Lexical Mismatch", ICASSP, 2005.

Huang, X, A. Acero, H-W Hon, "Spoken Language Processing - A Guide to Theory, Algorithm and System Development", Prentice Hall, 2001.

Kittler, J., Hatef, M., Duin, R., Matas, J., "On combining classifiers". IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (3), 226-239, 1998

Lit Ping Wong, Martin Russel, "Text-dependent speaker verification under noisy conditions using parallel model combination", The University of Birmingham, 2001.

"National institute of standard and technology. Speaker Recognition Evaluation Home Page" http://www.nist.gov/speech/test/spk/index.htm.

Moreno, A., Poich, D., Bonafonte,A ., Lleida, E. Llisterri, J., Mariño, J.B. and Nadeu, C., " ALBAYZIN speech database design of the phonetic corpus ", in *Proc. Eurrospeech,* 1993, pp. 175 - 178

Rabiner R., L., Schafer Ronald W. , "Digital Processing of Speech Signals", Prentice Hall,1975 .

Rabiner R., L., "A Tutorial On Hidden Markov Models And Selected Applications in Speech Recognition" , Proceedings of the IEEE, vol 77, No.2, February 1989, pp. 257 - 286

Ramos-Castro, D., Fierrez-Aguilar, J., Gonzalez-Rodriguez, J., Ortega-Garcia, J., "Speaker Verification Using Speaker And Test-Dependent Fast Score Normalization", Pattern Recognition Letters. Vol. 28(1), pp. 90-98, 2006.

Reynolds, D. A., Quatieri, T. F., Dunn, R. B., "Speaker Verification Using Adapted Gaussian Mixture Models", Academic Press, 2000.

Sam Kwong, Qianhua He, Y. K. Chan, "HMM Adaptation Techniques in Training Framework", Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology, 2001, vol 1, pp 350-354.

Sturium, D. E., Reynolds, D. A., "Speaker Adaptive Cohort Selection for Tnorm in Text-Independent Speaker Verification", ICASSP, 2005.

The HTK Book (for HTK Version 3.2.1). http://htk.eng.cam.ac.uk/docs/docs.shtml

Toledano, D. , Esteve, C., Gonzalez-Rodriguez , J., Morales, N., Fernández Pozo, R. And Hernández Gómez, L." MLLR Adaptation for Text-Dependent Speaker Recognition With Limited Enrolment Data, 2007.

Verlinde, P., Chollet, G., Acheroy, M., "Multi-modal identity verification using expert fusion" . Information Fusion 1 (1), 17-33, 2000.

ANEXO: Adaptación MLLR con distinto número de clases de regresión

En este anexo presentamos los resultados de verificación de locutor dependiente de texto para el caso de utilizar adaptación MLLR con distinto número de clases de regresión. Estos resultados se representan en forma de curvas DET y aunque su ubicación lógica en el desarrollo de esta memoria hubiese sido el apartado 4.1.1.2.2, el espacio ocupado por las gráficas desaconsejaba su inserción en el texto principal, por lo que hemos decidido trasladar las gráficas a este anexo.

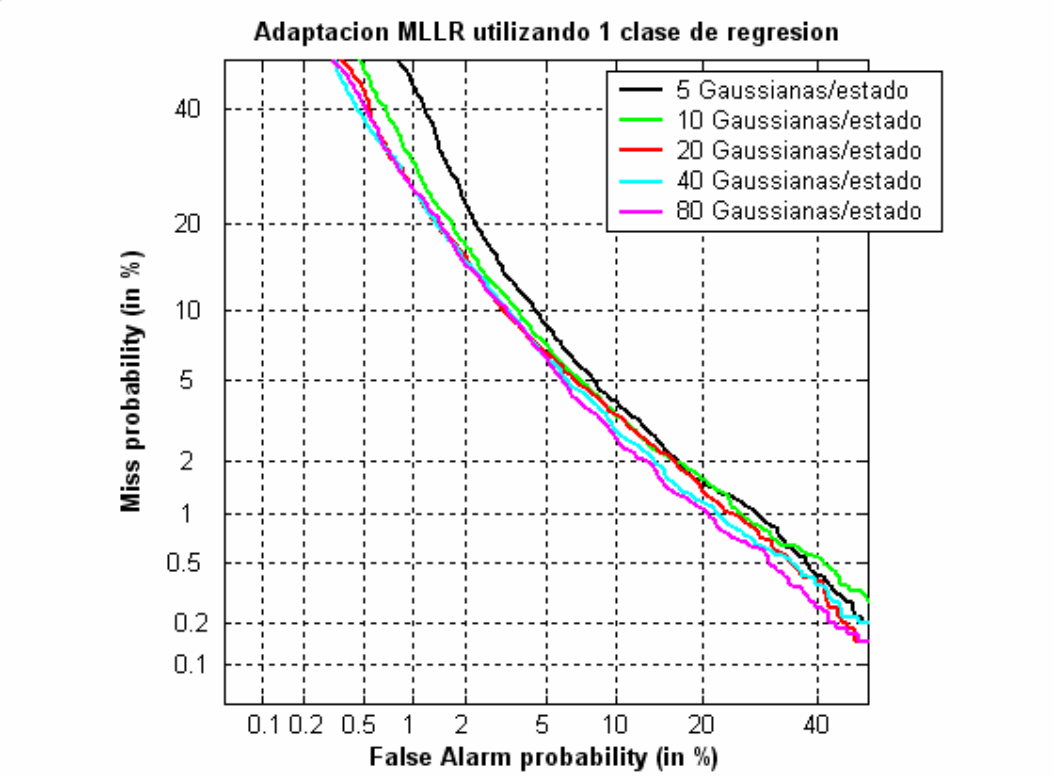

Figura 29. Curvas DET obtenidas sobre YOHO tras realizar adaptación MLLR con una clase de regresión en función del número de Gaussianas por estado

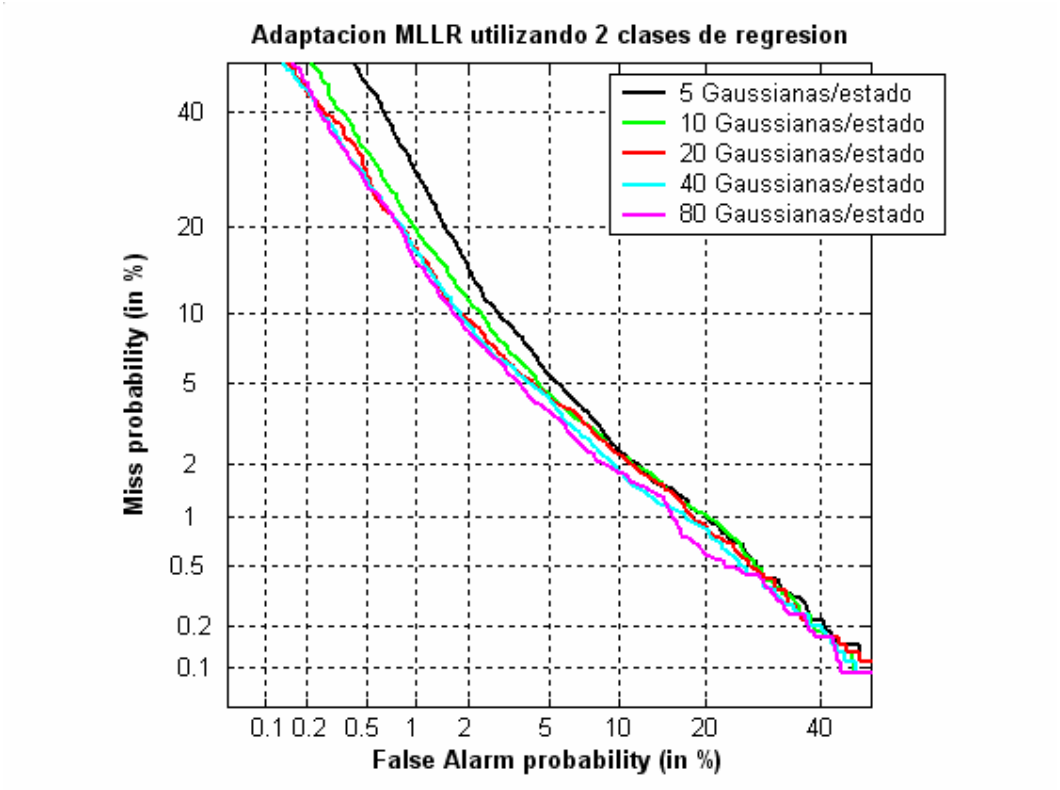

Figura 30. Curvas DET obtenidas sobre YOHO tras realizar adaptación MLLR con dos clases de regresión en función del número de Gaussianas por estado.

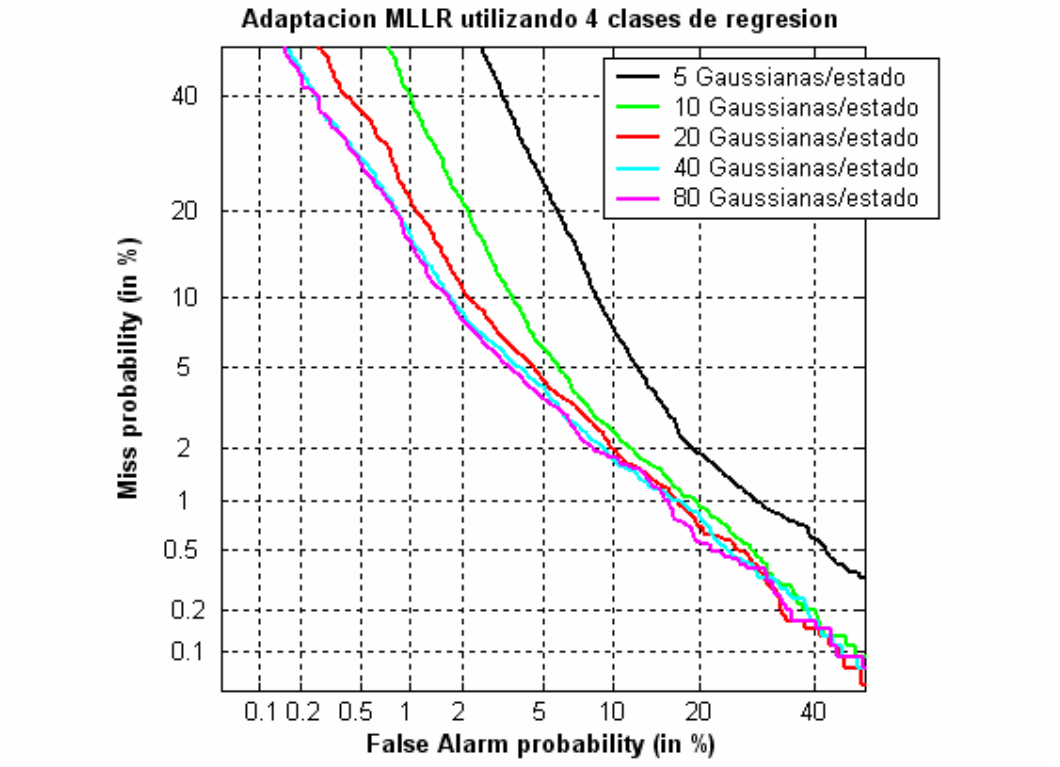

Figura 31. Curvas DET obtenidas sobre YOHO tras realizar adaptación MLLR con cuatro clases de regresión en función del número de Gaussianas por estado.

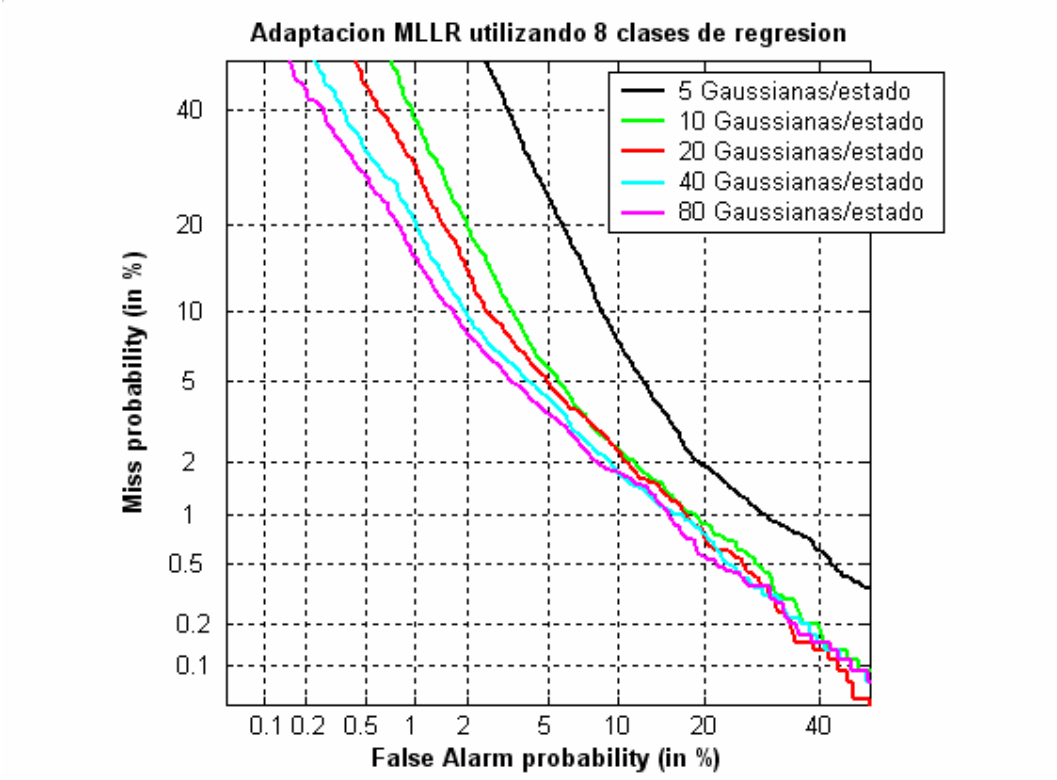

Figura 32. Curvas DET obtenidas sobre YOHO tras realizar adaptación MLLR con 8 clases de regresión en función del número de Gaussianas por estado.

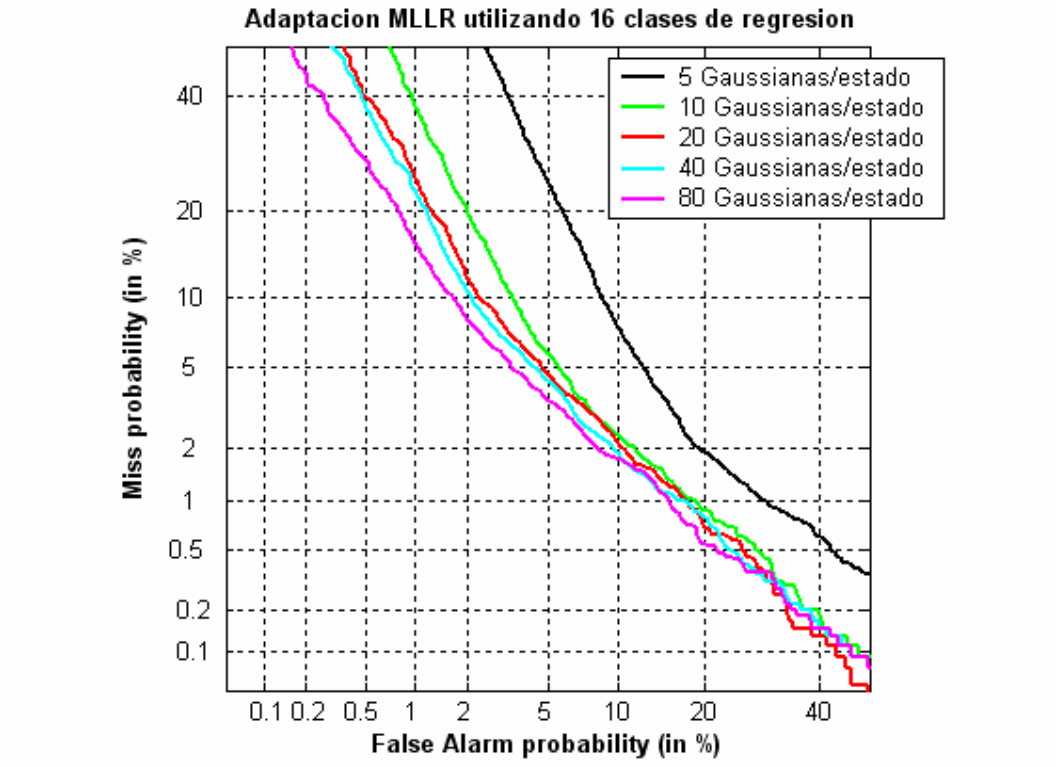

Figura 33. Curvas DET obtenidas sobre YOHO tras realizar adaptación MLLR con 16 clases de regresión en función del número Gaussianas por estado.

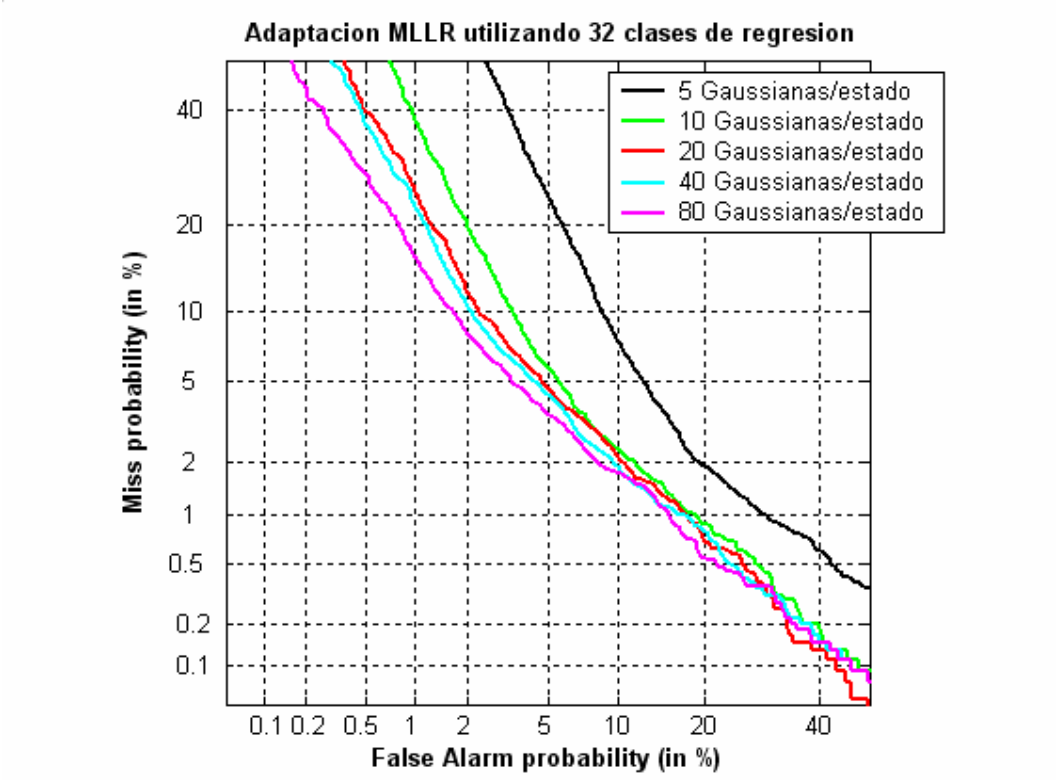

Figura 34.Curvas DET obtenidas sobre YOHO tras realizar Adaptación MLLR con 32 clases de regresión en función del número de Gaussianas por estado.

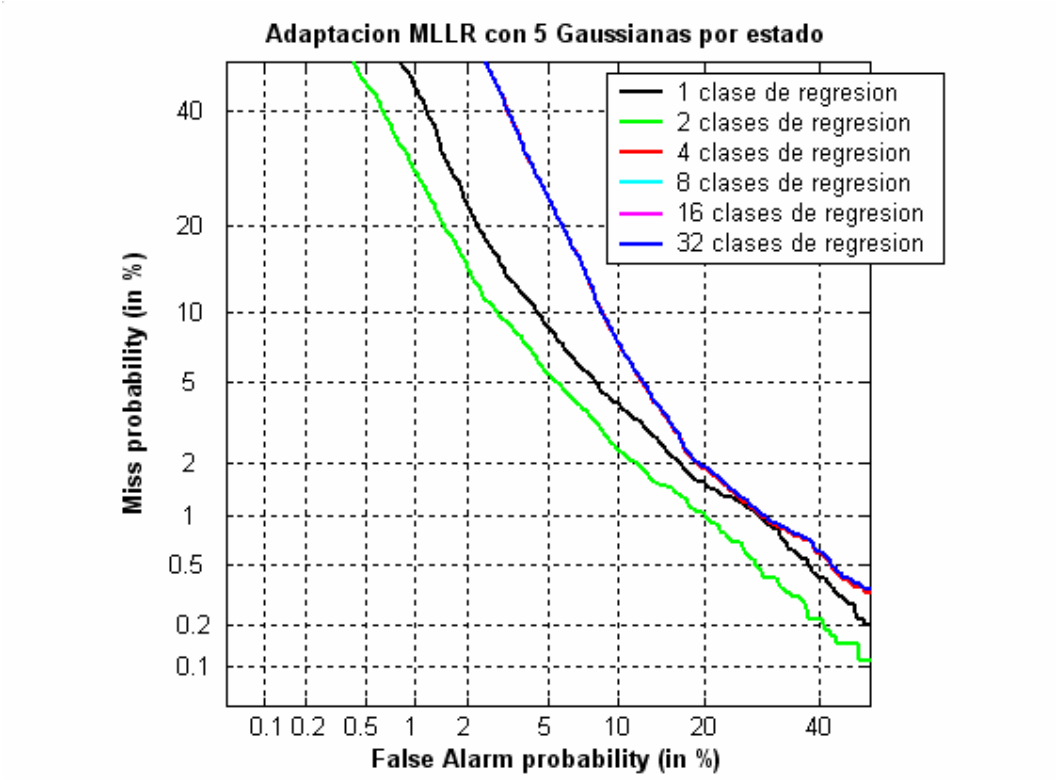

Figura 35. Curvas DET obtenidas sobre YOHO tras realizar adaptación MLLR con modelos de 5 Gaussianas por estado en función del número de clases de regresión.

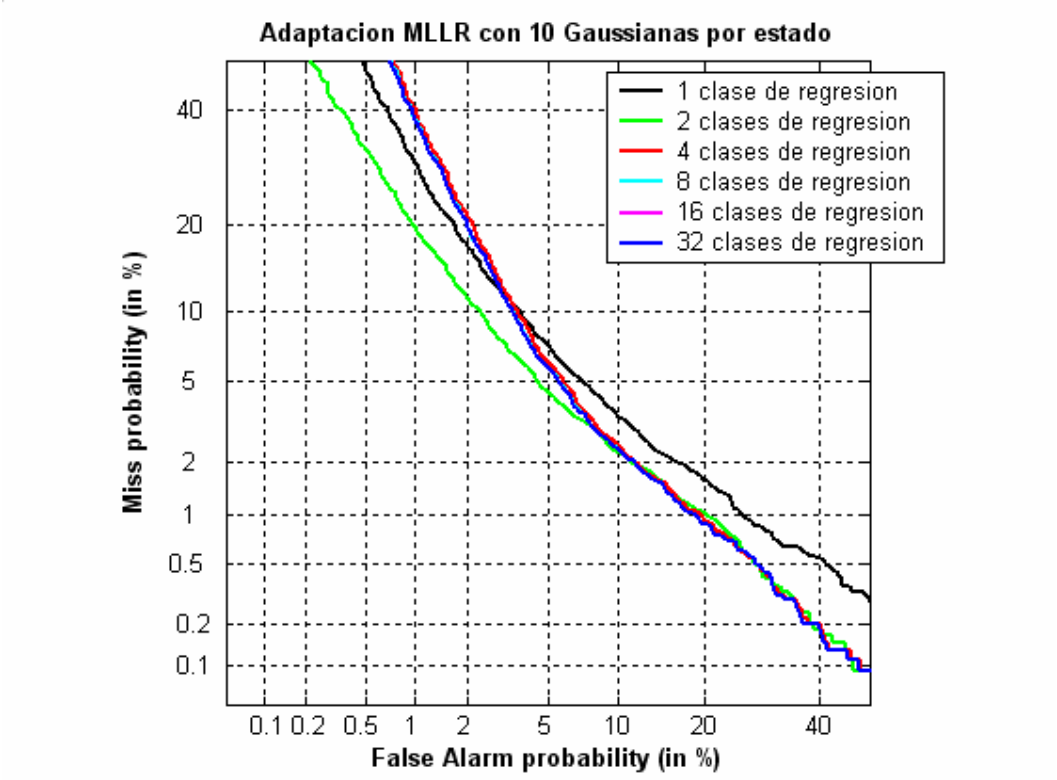

Figura 36. Curvas DET obtenidas sobre YOHO tras realizar adaptación MLLR con modelos de 10 Gaussianas por estado en función del número de clases de regresión.

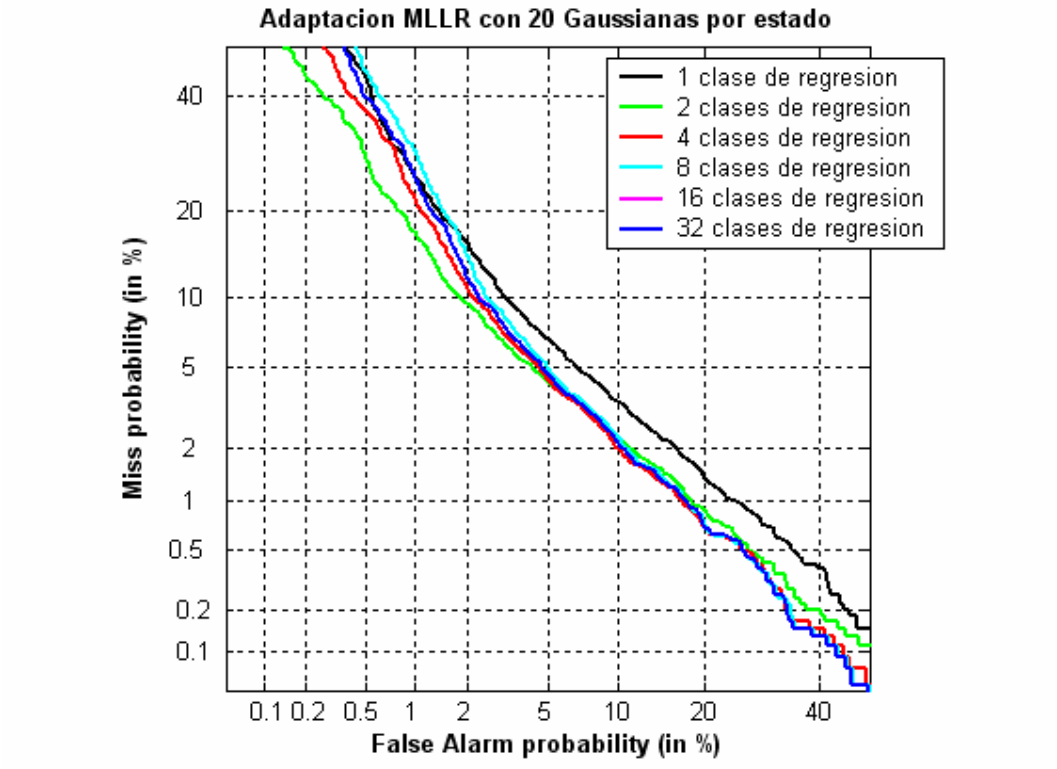

Figura 37. Curvas DET obtenidas sobre YOHO tras realizar adaptación MLLR con modelos de 20 Gaussianas por estado en función del número de clases de regresión.

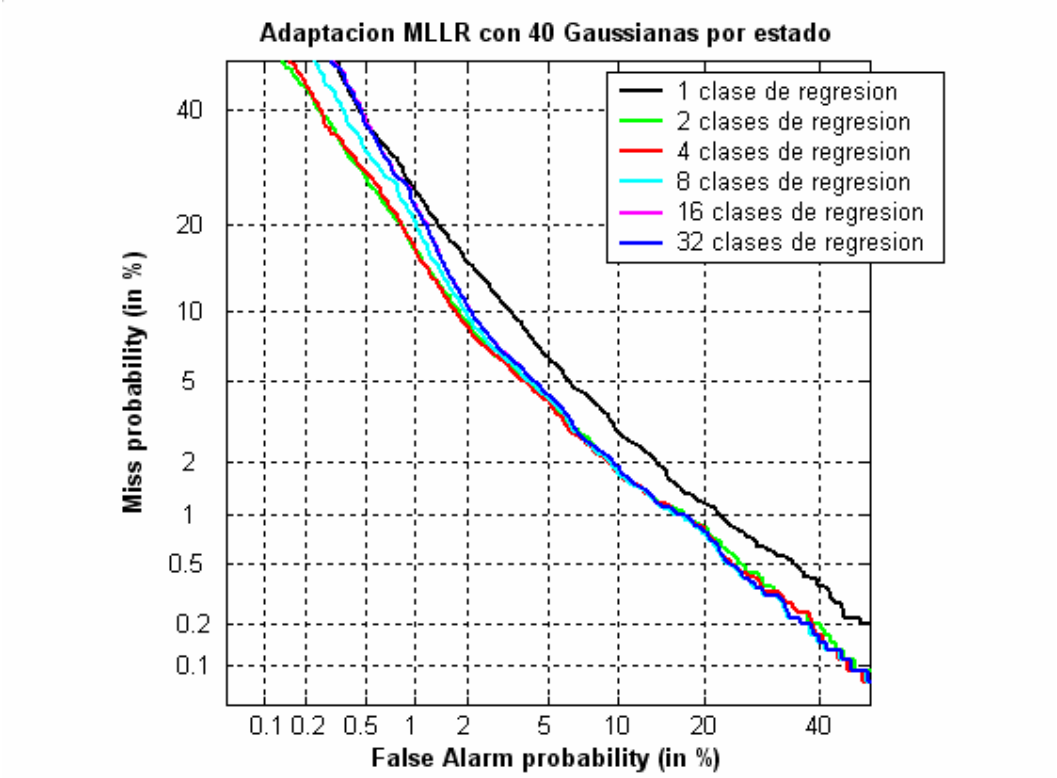

Figura 38. Curvas DET obtenidas sobre YOHO tras realizar adaptación MLLR con modelos de 40 Gaussianas por estado en función del número de clases de regresión.

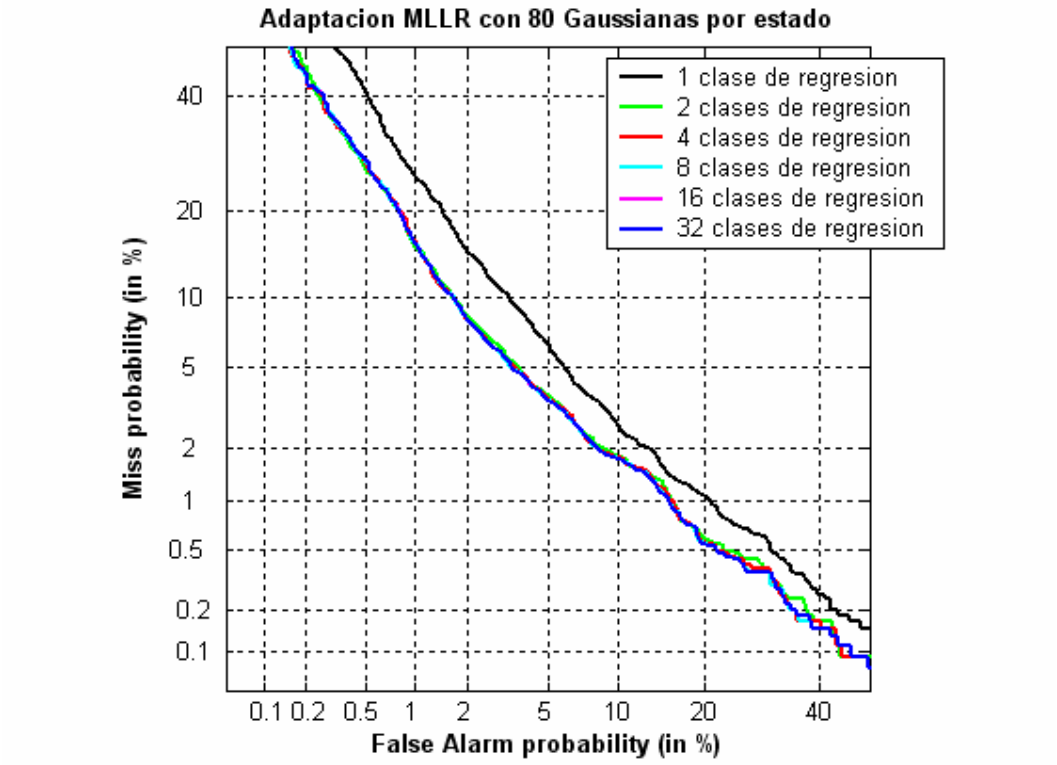

Figura 39. Curvas DET obtenidas sobre YOHO tras realizar adaptación MLLR con modelos de 80 Gaussianas por estado en función del número de clases de regresión.